NO. 2015-1632

United States Court of Appeals For the Federal Circuit

PHISON ELECTRONICS CORP.,

Patent Owner–Appellant,

v.

SILICON MOTION TECHNOLOGY CORP.,

Petitioner–Appellee.

Appeal from the United States Patent and Trademark Office,
Patent Trial and Appeal Board

PHISON ELECTRONICS CORP.'S OPENING BRIEF

David M. Barkan
Fish & Richardson P.C.
500 Arguello Street
Redwood City, CA 94063
(650) 839-5070

David Hoffman
Fish & Richardson P.C.
111 Congress Avenue, Suite 810
Austin, TX 78701
(512) 472-5070

Proshanto Mukherji
Fish & Richardson P.C.
One Marina Park Drive
Boston, MA 02210
(617) 542-5070

Attorneys for Patent Owner-Appellant

July 6, 2015

CERTIFICATE OF INTEREST

Counsel for Appellant Phison Electronics Corp. certifies the following:

1. The full name of every party I represent is: Phison Electronics Corporation.

2. The name of the real party in interest (if the party named in the caption is not the real party in interest) I represent is: N/A

3. All parent corporations of the party I represent are: N/A

4. All publicly held companies that own 10 percent or more of the stock of the party I represent are: Toshiba Corporation.

5. The names of all law firms and the partners or associates that appeared for the party now represented by me in the agency or are expected to appear in this court are:

Fish & Richardson P.C.: David M. Barkan, David M. Hoffman, Joshua A. Griswold, Proshanto Mukherji.

Dated: July 6, 2015

/s/ David M. Barkan
David M. Barkan

TABLE OF CONTENTS

Page

Certificate of Interest i

Statement of Related Cases vi

Statement of Jurisdiction vii

Statement of the Issues viii

Introduction 1

Statement of the Facts 3

I. The '267 Patent 3

A. Flash Memory Systems 3

B. The Garbled Code Problem 5

C. The '267 Patent's Invention 5

D. Representative Claim 9

II. The Sharon and Bennett References 10

III. The Proceedings Below 12

A. Silicon Motion Requests *Inter Partes* Review 12

B. The Board Institutes *Inter Partes* Review 13

C. Silicon Motion Moves for Reconsideration and is Denied 14

D. The Board Enters a Final Decision Holding the Claims at Issue Obvious Over Sharon and Bennett 15

Summary of the Argument 16

Legal Standards 17

TABLE OF CONTENTS (continued)

Page

Argument 18

I. The Board Legally Erred by Construing the Claims at Issue to Conflate an Indicator Bit, Which Indicates the Need for Predetermined or Replacement Data, With the Predetermined Data Itself 18

A. The Board's Construction is Inconsistent with the Claim Language and the Specification 19

B. The Board's Construction Is Also Erroneous Because it Eliminates a Feature of the Claimed Invention: Solving The Garbled Code Problem Within The Flash Memory System 21

II. The Board Erred by Relying on *KSR* to Expand the Specification 25

III. The Board's Obviousness Analysis is Wrong on the Law and the Facts 26

A. The Board Erred by Finding it Obvious to Import the Indicator Bit Into the Sharon System Because this Modification Would Make Parts of the Sharon System Superfluous 26

B. Sharon Teaches Away From Sending Bennett's Indicator Bit to the Host 29

Conclusion 32

TABLE OF AUTHORITIES

Page(s)

Cases

Alloc, Inc. v. ITC,
342 F.3d 1361 (Fed. Cir. 2003) 21

Bd. of Regents of the Univ. of Texas Sys. v. BENQ Am. Corp.,
533 F.3d 1362 (Fed. Cir. 2008) 19

C.R. Bard, Inc. v. U.S. Surgical Corp.,
388 F.3d 858 (Fed. Cir. 2004) 21

Consol. Edison Co. v. N.L.R.B.,
305 U.S. 197 (1938) 18

In re Abbott Diabetes Care Inc.,
696 F.3d 1142 (Fed. Cir. 2012) 21

In re Constr. Equip. Co.,
665 F.3d 1254 (Fed. Cir. 2011) 18

In re Cuozzo Speed Technologies, LLC,
778 F.3d 1271 (Fed. Cir. 2015) 18

In re Gartside,
203 F.3d 1305 (Fed. Cir. 2000) 18

In re Gurley,
27 F.3d 551 (Fed. Cir. 1994) 30, 31

Honeywell Int'l, Inc. v. United States,
609 F.3d 1292 (Fed. Cir. 2010) 24

ICU Med., Inc. v. Alaris Med. Sys., Inc.,
558 F.3d 1368 (Fed. Cir. 2009) 21

KSR Int'l Co. v. Teleflex Inc.,
550 U.S. 398 (2007) ..... *passim*

Linear Tech. Corp. v. Int'l Trade Comm'n,
566 F.3d 1049 (Fed. Cir. 2009) 19

TABLE OF AUTHORITIES (continued)

Page(s)

Microsoft Corp. v. Proxyconn, Inc.,
2015 WL 3747257 (Fed. Cir. June 16, 2015) 22

In re NTP,
654 F.3d 1279 (Fed. Cir. 2011) .....*passim*

Nystrom v. Trex Co.,
424 F.3d 1136 (Fed. Cir. 2005) 20

Phillips v. AWH Corp.,
415 F.3d 1303 (Fed. Cir. 2005) 19

Shire Dev., LLC v. Watson Pharm., Inc.,
2015 WL 3483245 (Fed. Cir. June 3, 2015) 24

St. Clair Intellectual Prop. Consultants, Inc. v. Canon Inc.,
412 F. App'x 270 (Fed. Cir. 2011) 20

In re Suitco Surface, Inc.,
603 F.3d 1255 (Fed. Cir. 2010) 1

Tech. Patents LLC v. T-Mobile (UK) Ltd.,
700 F.3d 482 (Fed. Cir. 2012) 24, 25

Teva Pharmaceuticals U.S.A., Inc. v. Sandoz, Inc.,
135 S.Ct. 831 (2015) 24

U.S. v. Adams,
383 U.S. 39 (1966) 30

UPI Semiconductor Corp. v. Int'l Trade Comm'n,
767 F.3d 1372 (Fed. Cir. 2014) 18

Other Authorities

37 C.F.R. § 42.104(b)(3) 12

STATEMENT OF RELATED CASES

No other appeal in or from the proceeding below has been brought in this or any other appellate court. Phison has sued Silicon Motion's customer, PNY Technologies, Inc., for infringement of the '267 patent and other patents in the District of Delaware in *Phison Electronics Corp. v. PNY Technologies Inc.*, No. 1-12-cv-01478 (D. Del.). That case is stayed pending the outcome of this appeal.

STATEMENT OF JURISDICTION

The Patent Trial and Appeal Board ("the Board") had jurisdiction over Silicon Motion's petition under 35 U.S.C. § 6. The Board issued a final written decision on January 28, 2015. A1. Phison timely filed its notice of appeal on March 31, 2015. A1286-1292. This Court has jurisdiction under 28 U.S.C. § 1295(a)(4)(A) and 35 U.S.C. § 141(c).

STATEMENT OF THE ISSUES

1. Whether the Board erred in construing the claims when it improperly broadened the meaning of "predetermined data" and conflated an indicator bit that indicates the need for predetermined data with the predetermined data itself;
2. Whether the Board erred in construing "predetermined data" by relying on *KSR Int'l Co. v. Teleflex Inc.* to expand the scope of intrinsic claim construction evidence to include both the patent specification and technology believed to be obvious in view of the patent specification;
3. Whether the Board erred when it held that it would be obvious to add an indicator bit feature to the system disclosed in the Sharon reference when that addition would make other parts of the Sharon system superfluous; and
4. Whether the Board erred when it concluded that Sharon does not teach away from the claimed features.

INTRODUCTION

This is an appeal from a decision in which the PTAB applied an overbroad claim construction and a flawed obviousness analysis. As this Court has repeatedly noted, the broadest reasonable construction rubric "does not give the PTO unfettered license to interpret claims to embrace anything reasonably related to the claimed invention." *In re Suitco Surface, Inc.,* 603 F.3d 1255, 1260 (Fed. Cir. 2010). Yet the Board here construed "predetermined data" in a manner that lacks support in the specification and conflicts with the '267 patent's description of "the invention."

The challenged '267 patent describes an innovative technique by which a flash memory system can avoid sending garbled and meaningless data to a host computer when it reads new memory blocks. When the flash memory system detects an indicator of a new block of data, the system sends an *equivalent amount* of non-garbled "predetermined data" in the place of the requested data that would be garbled if it was simply read from memory. The specification and claims repeatedly distinguish the indicator, which indicates a *need* for replacement, from the predetermined data that *is* the replacement.

Notwithstanding these plain teachings, the Board construed the claims to permit a new block indicator to be a form of predetermined data, thus conflating these distinct elements. The Board's construction rests on two errors in law. First, it finds no support in the specification, which distinguishes new block indicators from predetermined data. And second, the Board improperly relied on *KSR v. Teleflex* to

expand the scope of the intrinsic evidence to include not only the description in the specification but also other solutions that "cur[e] obvious problems [that] would be within the realm of common sense." A12. In doing so, the Board committed legal error by applying principles of obviousness to a question of claim construction.

The Board also erred as a matter of law by finding that it would have been obvious to add a second new block indicator to the system of its primary reference (Sharon) even though such an addition would be superfluous to Sharon's existing disclosure of an indicator.

Lastly the Board made a mistake of fact when it concluded without substantial supporting evidence that Sharon does not teach away from sending a one-bit indicator to a host in place of data. The Board improperly dismissed Sharon's clearly stated requirement that the memory system send the host uncompressed and non-abbreviated data – firmly ruling out any reason for a skilled artisan to modify Sharon to send the host a single bit of data in place of over 4,000 (as the Board's combination would require). A142 ¶ 43.

For each of these reasons, this Court should reverse or vacate the Board's obviousness ruling.

STATEMENT OF THE FACTS

I. The '267 Patent

The '267 patent[1] describes a solution to the "garbled code problem" that arises in flash memory systems. It describes a novel way for these systems to prevent sending garbled code to a host (*e.g.*, the computer they are connected to) and instead send predetermined data that is not garbled. A121 at 2:6-10.

A. Flash Memory Systems

A flash memory storage system uses solid state chip memory to store large amounts of data without needing a constant source of electric power. Consumers typically encounter them as USB flash drives or memory cards for cameras and the like. A123 at 5:22-25. A flash storage system connects to, and works together with a host, such as a computer, which reads and writes the data in it. *Id.* at 5:19-21. Fig. 1 of the '267 patent below shows a flash memory system's general organization:

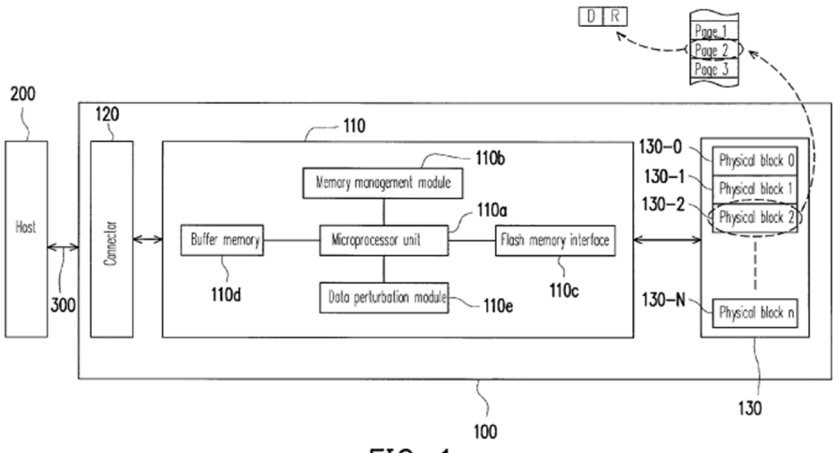


FIG. 1

[1] U.S. Patent 8,176,267.

A113. The storage portion of the flash memory is divided into "blocks" (elements 130-0 – 130-N), which are themselves sub-divided into smaller units called "pages." Pages in turn are divided into two parts: a "user data area D" that usually contains 512 bytes of user data[2], A124 at 7:5-9, and a small, separate "redundant area R" that stores control information such as "error correcting code[s]." *Id.* at 7:1-5. The '267 patent refers to this control information as "system data." *Id.*

The '267 patent explains that the flash memory system host cannot read or write individual bits, but rather only reads larger chunks of data. Usually, a "page [of data] is served as the *smallest* unit for reading and writing data," A123 at 6:67-7:1 (emphasis ours); but in some flash memory systems with larger page sizes (thousands of bytes), the page may be subdivided into a "plurality of sectors" (each being at least 512 bytes) and the sector may be the smallest programming unit. A123 at 6:55-7:1. This is because the size of the sector/page originates with the data sizes used in hard disk drives, which have a "user data area of 512 bytes and [a] redundant area of 16 bytes" for total size of 528 bytes.[3] A124 at 7:6-10. Often the terms "page" and "sector" are synonymous. *Id.* (explaining that sometimes "a page is a sector"). The disclosed embodiments all read data at least 512 bytes at a time. *Id.* at 6:55-7:10.

[2] 512 bytes of data consist of 4096 individual bits (ones or zeros).

[3] Sectors/pages are typically referred to as being 512 bytes in size because that is the amount of user data that is stored within them.

B. The Garbled Code Problem

The garbled code problem arises in flash memory systems that have what is called a "data perturbation" module. A121 at 1:47–50. For security purposes, this module encodes data as it is written to the flash memory. A123 at 6:14–21. It reverses the process when the data needs to be read. *Id.* However, the data perturbation module is only engaged when data is read from or written to the flash memory, not when the flash memory is first initialized—for example, at the time of manufacture. A121 at 1:60-62. Thus, a flash memory device can contain two types of memory blocks at any given time: (1) "new blocks" that still contain the *unencoded* data that was put in them during the initialization process, and (2) other blocks that contain *encoded* data that was subsequently written to them using the data perturbation module. A124 at 8:4-30; A123 at 6:14-42.

The problem arises when a host tries to read a page from a "new block" that has not been written to since the memory was initialized. Since the page of data was never encoded (as the block is new), the decoding process produces "garbled code" when the flash memory tries to decode it. A121 at 1:62-67. If the host receives this garbled data, it can cause errors and/or system crashes.

C. The '267 Patent's Invention

The '267 patent provides a novel and elegant solution to the garbled code problem. *Id.* at 2:6-10. When a flash memory system determines that the page to be read is located in a new block, and thus would cause garbled code to be returned to a

host, the flash memory system sends a page of predetermined data to take the place of the requested page of data. This solution is set forth in an unambiguous description of "the present invention." A126 at 11:60-67. As the '267 patent describes, "whether a physical block is a new block is determined before a data is transmitted…, and if the physical block is a new block, data 0x00 is transmitted to the end-user system by changing the page or physical block to be read or directly producing the data 0x00." *Id.*

The patented solution has three significant aspects. First, there is the predetermined data itself that is returned to the host in replace of the garbled data. Second and separately, there is information that indicates to the system that such a replacement should be done. This distinction between the predetermined data and the indicator that it should be used is significant to the invention of the '267 patent and every embodiment within the '267 patent preserves and relies upon this distinction. Information that indicates that a replacement should happen is not the data that is returned to the host. Third, the problem of garbled data is solved within the flash memory storage system itself. The host that requested the garbled data receives predetermined data and need not even be aware that any substitution was made.

These aspects come through in each of the three embodiments in the '267 patent, which are referred to as the "first," "second," and "third" embodiments. A124 at 8:33; A125 at 9:63; A126 at 11:5. The three embodiments differ according to (1) how the system detects whether the page being read is part of a new block, and (2)

how it obtains the predetermined data that it sends to the host in place of the garbled data. However, in all of the embodiments, the new block indicator is district from "predetermined data." And in all of the embodiments, the flash memory storage system itself solves the problem of garbled code. The embodiments are summarized in the following table:

	How it detects a "new block"	**How it obtains replacement data**
First Embodiment	Checks an indicator bit stored in the redundant area of the first page of each block. (A125 at 9:20-22)	Reads a page of predetermined (*i.e.,* correct) data from a different page in the flash memory block. (A125 at 9:1–4.)
Second Embodiment	Checks an indicator bit stored in the logical-physical mapping table. (A125 at 10:7-12)	Reads a page of predetermined (*i.e.,* correct) data from a page in a different flash memory block. (A125 at 10:28-32.)
Third Embodiment	Reads the garbled data and checks if it matches the data that would be produced by decoding the contents of a new block. (A126 at 11:14–23.)	Changes the data into a page of predetermined (*i.e.,* correct) data. (A126 at 11:51-54.)

The first embodiment maintains an indicator bit in the redundant area of the first page of each block, which flags whether the block is a new block or not. A125 at 9:20-22. ("[T]he indicator indicating whether a physical block is a new block is recorded in the redundant area R of the first page in [each] physical block."). During the card activation process, it fills the data area of the first page of each block with correct (encoded) data. A124 at 8:44–49. Thereafter, when it receives a request from

the host to read a page of data, "[i]f the page to be read is in a new block, [it] reads the first page of the physical block and transmits the data therein to the host ..." A125 at 9:1-4. Notably, the indicator is distinct from the "predetermined data," is stored only in the redundant area, and is not returned to the host as predetermined data.

The second embodiment maintains a similar new-block indicator flag but stores it in the logical-physical mapping table, which is separate from the "predetermined data." *Id.* at 10:7–12. Instead of generating one page of correct data in each block during card activation, the second embodiment selects one entire block and fills all of its pages with correct, non-garbled data. *Id.* at 10:17–21. Then when it receives a request for a page of data, "if the physical block corresponding to the page to be read is a new block, [it] reads data from the corresponding correct page in the alternate physical block and transmits the data to the host." *Id.* at 10:28-32. The indicator again is not returned to the host.

The third and final embodiment does not rely on the new block indicator. Instead, it reads the data from each requested page, A126 at 11:14–23, and determines whether the page is part of a new block by comparing what it read with the garbled data that would have been produced if the page were in a new block. *Id.* If the block is determined to be a new block, the flash memory system *changes* the data to the correct replacement data (*e.g.,* a page worth of zeroes represented by the shorthand 0x00) and sends that data to the host. *Id.* at 11:51-54 ("[T]he data read from the physical block is *changed to* 0x00, and the data 0x00 is transmitted to the host 200.") (emphasis added).

As shown above, a common thread in each of the embodiments is the fact that the flash memory system implements the entire solution to the garbled data problem without involving the host. The claims and specification also define this as a feature of the invention. For example, independent claims 1, 11, and 22 recite a method for returning data to a host, an improved flash memory controller, and an improved flash memory storage system, respectively. No aspect of these claims take place on a host. Like the claims, the specification explains that the "present invention" relates to a "data accessing method which can prevent a host from reading garbled codes, and a flash memory storage system and controller using the same." A121 at 1:19–20; *see also id.* at 2:7–22, and that the flash system's controller is what "executes [the] data accessing method …" A123 at 5:50–52; s*ee also* A121 at 2:23–35.

D. Representative Claim

Claim 1 of the '267 patent is representative of the claims at issue:

> 1. A data accessing method, suitable for a flash memory storage device having a data perturbation module, wherein a flash memory of the flash memory storage device has a plurality of physical blocks, and the physical blocks are grouped into at least a data area and a spare area, the data accessing method comprising:
>
> receiving a read command from a host, and obtaining a logical block to be read and a page to be read from the read command;
>
> determining whether a physical block in the data area corresponding to the logical block to be read is a new block;
>
> transmitting a predetermined data to the host when the physical block corresponding to the logical block to be read is the new block; and

> decoding data read from the physical block corresponding to the logical block to be read by the data perturbation and transmitting the decoded data to the host when the physical block corresponding to the logical block to be read is not the new block.

II. The Sharon and Bennett References

The Board held certain claims of the '267 patent obvious over a combination of US 2008/0151618 to Sharon et al. ("Sharon") and US 2007/0113030 to Bennett et al. ("Bennett"). Sharon does not address the garbled-code problem. Rather, it is about randomizing data in devices where non-random data patterns can cause errors. A143 ¶ 62. Sharon explains that certain flash devices—"multi-bit cell" ones, which store more than one bit in each flash memory cell, A138 ¶ 6—are subject to unpredictable errors when they store non-random data. A140 ¶¶ 21–22. Sharon describes a data-storage technique that "randomiz[es] original data while preserving a size of the original data" and, "in response to a request for the original data: retriev[es] the randomized data from the memory, derandomiz[es it], … and export[s] the retrieved data to" the host. *E.g.* A142 ¶ 39 (internal numbering omitted). Importantly, Sharon distinguishes its invention from prior art specifically because Sharon ensures that "the size of the original data is preserved" when storing and retrieving data, rather than being changed by data compression or other methods. *Id.* ¶ 43. In Sharon, returning exactly what was requested is paramount. *Id.*

Although it does not address the garbled-code problem, Sharon mentions in passing that its technique might cause "confusion" between a flash memory page that

has been erased, which it fills with a sequence of "1" bits, and one that was actually programmed with such a sequence:

> When a flash page is erased all of the cells of the page … are assumed to contain the fixed all-1's data pattern. This might cause a confusion with a page that was actually programmed to the all-1's bit sequence, but that, according to the present invention, represents some other data bit sequence.

A145 ¶ 78. Sharon explains that this confusion can be eliminated by using a "flag cell" on each page to record whether the page is written or unwritten, or by other well-known methods:

> However, this [confusion] can be handled by the application using the flash memory device being able to distinguish a page that was not written yet from a page that was written. This is easy to do and is well known in the prior art of flash management systems, for example by allocating one or more *flag cells*, in the management portion of a page, that are always written as part of the page programming operation, and thus if found to be in the leftmost state, indicate an unwritten page. So a page found to be unwritten is interpreted according to the standard prior art logic, while a page found to be written is interpreted according to the methods of the present invention.

Ex. 1002 ¶ 0078 (emphasis ours). As the Board found, Sharon does not disclose sending any "predetermined" or "replacement" data to the host if an unwritten block is to be read. A10.

Bennett similarly addresses neither the '267 patent's garbled-code problem nor Sharon's randomization problem. Rather, as the Board found, "Bennett is directed to methods of managing erase operations" in flash memories. A10 (citing A150 (Bennett Abstract)). Bennett has an "Erased Status" indicator bit that does the same thing as

Sharon's "flag cell": it identifies unwritten pages. *Id.* (citing A171 ¶ 153). Bennett also discloses that this indicator bit could be sent to the host as part of read operation. *Id.* (citing A171 ¶¶ 155, 161).

As discussed below, the Board concluded that it would be obvious to modify the Sharon system in two steps in light of Bennett: first, by adding in the "Erased Status" indicator bit to Sharon (on top of the "flag cells" already present); and then, by transmitting this bit to the host instead of the data requested by the host. A17–18. This was an error.

III. The Proceedings Below

A. Silicon Motion Requests *Inter Partes* Review

Phison sued Silicon Motion's customer, PNY, for infringement of the '267 patent in district court on November 15, 2012. About eight months later—on July 30, 2013—Silicon Motion filed a petition for *inter partes* review of the '267 patent. A111. Silicon Motion's petition challenged all the claims of the '267 patent on a variety of overlapping grounds. A54. It asserted five grounds for invalidity, among which were allegations of anticipation and obviousness over a number of different references and combinations of references. *Id.*

Silicon Motion's petition did not state "[h]ow the challenged claim is to be construed" as Patent Office regulations require, *see* 37 C.F.R. § 42.104(b)(3); A55, and indeed it disavowed all claim constructions. *See* A55 ("[I]nterpretation or construction, or both, of the claims in the '267 patent relevant to this inter partes review should not

be viewed as constituting…petitioner's own interpretation or construction for any other purpose ...").

Phison filed a timely preliminary response in which it pointed out this deficiency in Silicon Motion's petition and argued that it was grounds to reject the petition. A453–454. In the alternative, Phison responded to Silicon Motion's arguments on the merits. A459–484. In the course of doing so, Phison argued that the term "predetermined data," in the context of the '267 patent, should be construed as "replacement data stored before user operation." A454–459.

B. The Board Institutes *Inter Partes* Review

The Board addressed these issues in its Institution Decision. A24–43. Instead of rejecting Silicon Motion's petition for failure to provide a claim construction, the Board in its discretion found that Silicon Motion had merely waived the right to propose a specific claim construction for any term. A31 ("Silicon Motion may gain no benefit that it might have had from a more specific construction ….). With that caveat, the Board considered the merits.

The Board began by construing "predetermined data" to mean "replacement data,"[4] A29, so as to be "consistent with examples of use from the Specification." *Id.* It then rejected four of Silicon Motion's five grounds for institution, A31–33, A39–40,

[4] The Board also determined that "predetermined data" did not have to be "stored before user operation," as Phison had previously argued. A29. Phison does not challenge this aspect of the Board's ruling.

but instituted *inter partes* review of some claims on the remaining one. A38. Specifically, it instituted review of claims 1, 3-7, 9-11, 13-17, 21, 22, 24, and 25 ("the claims at issue") as obvious over a combination of Sharon and Bennett. A41.

The Board found that Sharon, standing alone, did not disclose any specific solution to the garbled code problem, and particularly did not disclose the key step of "send[ing] predetermined data in place of read data for new blocks." A34. Sharon discloses only that some unspecified "standard prior art logic" can be applied when the host tries to read an unwritten block, A145 ¶ 78 ("[A] page found to be unwritten is interpreted according to the standard prior art logic …"), and the Board found that it would be "mere speculation" to read into this the '267 patent's particular solution that uses "predetermined data," which it had construed as "replacement data." A33.

But the Board then erroneously found that Bennett could fill in these gaps. The Board reasoned that it would be obvious to modify Sharon by (1) adding in Bennett's "Erased Status" indicator bit that tracks unwritten blocks (in addition to the existing "flag cell" that does the same thing) and then (2) sending Bennett's indicator bit to the host when it tried to read data from a new block. *See* A34–35. This, the Board held, might invalidate the claims at issue. *Id.*

C. Silicon Motion Moves for Reconsideration and is Denied

Silicon Motion moved for partial reconsideration of the Board's Institution Decision. *See* A541. It belatedly urged the Board to adopt a different construction of "predetermined data," A549–550, and also argued that Bennett in fact disclosed not

only sending a single indicator bit to the host but also sending it actual corrected data. A550–551. The Board found that Silicon Motion had waived both arguments by not presenting them in its Petition. A637 (claim construction argument); A639 (argument regarding Bennett).

D. The Board Enters a Final Decision Holding the Claims at Issue Obvious Over Sharon and Bennett

The Board entered a Final Written Decision on January 28, 2015, after receiving additional briefs from the parties and conducting a hearing. A1–A23. The Board adhered to its earlier conclusions. Once again, it found that in light of Bennett, it would be obvious to modify Sharon by first adding in an indicator bit and then by sending that bit to the host when it tried to read data from a new block. *See* A17–18. It found the claims at issue obvious as a result. A18.

The Board construed the claims broadly. Although it acknowledged that "the 'replacement data' must necessarily replace the decoded data," it nevertheless held that "[n]othing in claim 1, or the other independent claims ... require[s] that the 'replacement data' [sent to the host] must be the same or similar to the decoded data such that the host thinks that it received the requested data." A12. Accordingly, it held that the "predetermined" or "replacement" data of the claims could be a single bit. *See id.*

The Board rejected Phison's arguments against obviousness. Phison had pointed out that Sharon already discloses a "flag cell" that does exactly what Bennett's

indicator bit does. A592 (citing A145 ¶ 78). The Board did not dispute this. A16. But the Board nevertheless held that it would have been obvious to import a second such identifier—Bennett's indicator bit—into the Sharon system. *Id.*; A18.

Phison had also argued that Sharon teaches away from sending a one-bit response to the host in lieu of the requested data, and indeed lists its ability to return data of the same size requested as a distinguishing feature of its invention. A590–591 (citing A142 ¶ 43). The Board rejected this argument as well. A15–16.

Phison timely appealed. A1286–1290. Silicon Motion did not cross appeal.

SUMMARY OF THE ARGUMENT

The Board erred on four points. It first erred when it construed the asserted claims to "[not] require that the 'replacement data' [sent to the host] must be the same or similar to the decoded data such that the host thinks that it received the requested data," and it compounded that error by concluding that a one-bit new-block indicator was itself "replacement data." A12. The claims and specification provide a more limited definition of what the predetermined replacement data is and how it is used. They distinguish between the indicator bit, which indicates that "predetermined" or "replacement" data is needed because a new block is being read, and the predetermined data itself; they limit the invention to one that replaces garbled code by an equivalent amount of replacement data, rather than a single bit; and they provide that the invention must solve the garbled data problem within the flash memory device rather than by relying on the host to translate an indicator into the correct data.

The '267 patent's approach, unlike the Board's, also maintains compatibility with key industry standards.

Second, in coming up with this overbroad construction, the Board improperly relied on *KSR* to expand the intrinsic record to encompass not just what the patent teaches but also what the Board believed might be obvious in view of the specification. A12. This is wrong as a matter of law because it conflates claim construction with obviousness.

Third, the Board erred in its obviousness analysis. It made a mistake of law when it held that adding Bennett's indicator bit to Sharon's system would have been obvious, as that cannot be done without rendering part of Sharon's disclosure superfluous. *See In re NTP*, 654 F.3d 1279, 1298–99 (Fed. Cir. 2011). And it made a clear mistake of fact when it concluded that Sharon does not teach away from sending a one-bit response to the host. Sharon unequivocally does.

LEGAL STANDARDS

"The ultimate judgment of obviousness is a legal determination," *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 427 (2007), which is reviewed *de novo*. *In re Gartside*, 203 F.3d 1305, 1316 (Fed. Cir. 2000). Any underlying factual findings are reviewed for substantial evidence. *Id.* "The determination of what a reference teaches is one of fact, as is the existence of a reason for a person of ordinary skill to combine references." *In re Constr. Equip. Co.*, 665 F.3d 1254, 1255 (Fed. Cir. 2011).

The claim construction standard is similar. Again, this Court reviews "the ultimate construction of the claim *de novo,*" reviews factual findings made from intrinsic evidence *de novo*, and reviews the Board's "underlying factual determinations concerning extrinsic evidence for substantial evidence." *In re Cuozzo Speed Technologies, LLC*, 778 F.3d 1271, 1282-83 (Fed. Cir. 2015) (citing *Teva Pharmaceuticals U.S.A., Inc. v. Sandoz, Inc.,* 135 S.Ct. 831, 841 (2015)).

"Substantial evidence is more than a mere scintilla. It means such relevant evidence as a reasonable mind might accept as adequate to support a conclusion." *Consol. Edison Co. v. N.L.R.B.*, 305 U.S. 197, 229 (1938); *UPI Semiconductor Corp. v. Int'l Trade Comm'n,* 767 F.3d 1372, 1379 (Fed. Cir. 2014), *reh'g denied* (Dec. 1, 2014).

ARGUMENT

I. The Board Legally Erred by Construing the Claims at Issue to Conflate an Indicator Bit, Which Indicates the Need for Predetermined or Replacement Data, With the Predetermined Data Itself.

The Board legally erred when it conflated the indicator bit, which indicates that "predetermined data" is needed because a new block is being read, with the predetermined data itself. Such questions about "the proper scope of the asserted claims" are claim construction issues. *Bd. of Regents of the Univ. of Texas Sys. v. BENQ Am. Corp.*, 533 F.3d 1362, 1367 (Fed. Cir. 2008); *see also Linear Tech. Corp. v. Int'l Trade Comm'n*, 566 F.3d 1049, 1059 (Fed. Cir. 2009). The Board's construction was wrong for two reasons.

A. The Board's Construction is Inconsistent with the Claim Language and the Specification

The claims distinguish between a new-block indicator and the "predetermined data" (or "replacement data," in the words of the Board's construction). Many of the claims—for example, claim 3—recite the "predetermined data" and the "indicator" separately. *See* A126–127, claims 3, 13, 24. And every time, they expressly say that the "indicator" is what is used at the "determining" step to tell if a new block is being read, while the "predetermined data" is what is sent to the host in the "transmitting" step if so. *See id.*

The specification makes the same distinction. The "specification ... is the single best guide to the meaning of a disputed term," *Phillips v. AWH Corp.*, 415 F.3d 1303, 1315 (Fed. Cir. 2005) (en banc), and "may define claim terms by implication such that the meaning may be found in or ascertained by a reading of the patent documents." *Id.* at 1321 (quoting *Irdeto Access, Inc. v. Echostar Satellite Corp.,* 383 F.3d 1295, 1300 (Fed. Cir. 2004)). That is the case here.

First, both embodiments in the specification that maintain an indicator (i.e. the first and second embodiments) use it solely to determine if the requested block is a new block. A125 at 9:20-22, 10:7-12. They do *not* send it to the host in the "transmitting" step; instead, they obtain and send a full page of correct replacement data. *Id.* at 9:1–4, 10:28–32. And since a page contains 512 bytes (i.e. 4,096 bits) of user data, A124 at 7:6, this means that they go out of their way to send over four

thousand times more data than the indicator bit. This is inconsistent as a practical matter with the idea that the one-bit indicator is an adequate substitute.

Second, the specification repeatedly, consistently, and exclusively discloses that garbled code is replaced by an equivalent amount of predetermined data, rather than a one-bit indicator. When data is to be read from a new block: the first embodiment changes the page to be read to one where corrected data has previously been stored, "reads [that] page," and then "transmits the data therein to the host," A125 at 9:1–4; the second embodiment instead changes the block to be read to one with corrected data, and again "reads data from the corresponding page in the selected physical block and transmits the data to the host," *id.* at 10:28–32; and the third embodiment reads the garbled code and actually "change[s]" it to correct data before sending it to the host. A126 at 11:51–54. All send the host the amount of replacement data—one page—that the host requested. Such "consistent use of a claim term in the specification suggests that the scope of a claim is limited." *St. Clair Intellectual Prop. Consultants, Inc. v. Canon Inc.*, 412 F. App'x 270, 273 (Fed. Cir. 2011); *Nystrom v. Trex Co.*, 424 F.3d 1136, 1144-45 (Fed. Cir. 2005) (adopting narrow construction based on specification's "consistent[] use[of] the term"); *In re Abbott Diabetes Care Inc.*, 696 F.3d 1142, 1150 (Fed. Cir. 2012) (same based on specification's "repeated[], consistent[], and exclusive[]" narrow use of term in every embodiment).

Finally, lest any doubt remain, the specification expressly limits the "present invention" to using replacement data that is obtained according to one of the three

embodiments. In particular, it states that "according to the data accessing method provided by the *present invention*, data … is transmitted to the end-user system" in one of exactly two ways: either "by changing the page or physical block to be read," as in the first and second embodiments, or "[by] directly producing the data," as in the third embodiment. A126 at 11:59–65 (emphasis ours). It does not include substituting the new-block indicator. Such "[s]tatements that describe the invention as a whole … are more likely to support a limiting definition of a claim term." *C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 864 (Fed. Cir. 2004); *see also In re Abbott Diabetes Care Inc.*, 696 F.3d 1142, 1149 (Fed. Cir. 2012); *ICU Med., Inc. v. Alaris Med. Sys., Inc.*, 558 F.3d 1368, 1375 (Fed. Cir. 2009); *Alloc, Inc. v. ITC*, 342 F.3d 1361, 1368-70 (Fed. Cir. 2003).

In short, the Board conflated concepts that the patent differentiates, namely the indicator that flags the need for predetermined or replacement data and the predetermined data itself. This was error. The broadest reasonable construction standard is not a license for the Board to "construe the claims during IPR so broadly that its constructions are unreasonable under general claim construction principles." *Microsoft Corp. v. Proxyconn, Inc.*, 2015 WL3747257 at *6 (Fed. Cir. June 16, 2015).

B. The Board's Construction Is Also Erroneous Because it Eliminates a Feature of the Claimed Invention: Solving The Garbled Code Problem Within The Flash Memory System

The Board's construction is also overbroad because it requires the host to interpret the indicator bit and translate it into the corrected data. By contrast, the claimed invention solves the garbled code problem entirely within the flash memory

system. The claims and specification define this as a feature of the invention, and extrinsic evidence in the form of industry standards show that it is a critical one.

The specification requires the flash memory system to solve the garbled code problem by itself. It defines the "invention" as (1) a "data accessing method" that solves the garbled code problem, and (2) a flash memory system that uses this *entire* method:

> "The present invention … relates to … a data accessing method which can prevent a host from reading garbled codes, and a flash memory storage system and a controller using the same."

A121 at 1:19–20; *see also id.* at 2:7–22 (similar). Similarly, the specification provides that "the memory management module" in the flash device's controller is what "executes [the] data accessing method …." A123 at 5:50–52. Because the "data accessing method" solves the problem and that method is executed within the flash memory system, it follows that, as defined by the specification, the solution takes place in the flash memory system.

The claims confirm this. Method claim 1 defines the "data accessing method" as a process that begins when the flash memory system "receiv[es] a read command from a host" and ends when data is "transmitt[ed] ... to the host" in response. A126 claim 1; *see also* A121 at 2:23–35. And the apparatus claims are respectively directed to a "flash memory storage system" (claim 22 and dependents) and a "controller" component therein (claim 11 and dependents) that implement the method of claim 1.

Industry standards from around the filing date show that this feature is critical because flash memory systems need to interoperate with cameras and other hosts that cannot interpret the indicator bit. For instance, the contemporary Secure Digital Memory Card Standard ("SD Card Standard"), which specified how hosts communicated with "memory card[s]," A123 at 5:22-25, requires data to be transmitted back and forth in chunks, which it calls "blocks,"[5] rather than bit-by-bit. A192 ("Data transfers to/from the SD Memory Card are done in blocks."); A1075 at ll. 20–25 (Silicon Motion's expert conceding that "the [SD Card] specification does not define the block size to be as small as a bit."). Similarly, the Microsoft Windows SD card interface would generate an error if it received a single bit from a flash memory in response to a read request, as Silicon Motion's expert conceded. A1027 at ll. 80:7-13:

> Q So you would agree that the designers of Microsoft Windows chose to design their system such that returning less than one byte of data would cause an error?
>
> A Within the context of the operation of Microsoft Windows, Microsoft Windows would generate an error should such a situation occur.

The Board did not disagree with these facts. With regard to Windows, it said that it "may agree with [Phison's expert] that Sharon operating according to the Windows ReadFile function may return an error when a single bit is received." A15.

[5] These are not necessarily the same size as the "blocks" discussed in the '267 patent.

It made no factual finding about the SD Card Standard. Absent specific findings of fact supporting the Board's construction, it is reviewed without deference. *Teva Pharm. USA, Inc. v. Sandoz, Inc.*, 135 S. Ct. 831, 840-41 (2015); *Shire Dev., LLC v. Watson Pharm., Inc.*, No. 2013-1409, 2015 WL 3483245, at *1 (Fed. Cir. June 3, 2015).

In short, by sending four thousand times more data than the indicator bit, the invention makes it possible for the flash memory device to work properly with preexisting hosts that operate according to preexisting industry standards like the ones above. It does not require that either the standards or the hosts that implement them be modified to accept an indicator bit in response to a read request. By broadening the claims to encompass a flash memory system that only works when connected to a new kind of host, the Board ignored one of the fundamental features and advantages of the '267 patent's invention. *See*, *e.g.*, *Honeywell Int'l, Inc. v. United States*, 609 F.3d 1292, 1299 (Fed. Cir. 2010) (limiting claim term to use of "perceptible red light" on the ground that "there would be no point, in the context of this invention, to [use] … light that cannot be seen.") (emphasis ours); *Tech. Patents LLC v. T-Mobile (UK) Ltd.*, 700 F.3d 482, 493 (Fed. Cir. 2012) (same for proposed claim construction that "ignore[d] … the very purpose of the invention.").

Properly construed, therefore, the claimed invention solves the garbled code problem within the flash memory system. This means that, just as when a non-new block is read, the flash memory has to decode the data there and send that usable decoded data to the host, *E.g.* A126–127 (claim 1), so when a new block is read, the

flash memory must replace the garbled data in new blocks with directly usable predetermined data before sending it to the host.

II. The Board Erred by Relying on *KSR* to Expand the Specification

The Board improperly attempted to fill the gap between the specification and its construction by relying on *KSR*. A12. It held that *KSR* allowed it to base its construction of "predetermined data," not only on the intrinsic and extrinsic evidence, but also on other technologies that were within the realm of "common sense":

> It is important to remember that "in many cases a person of ordinary skill will be able to fit the teachings of multiple patents together like pieces of a puzzle," *KSR Int'l Co.*, 550 U.S. at 420, such that curing obvious problems would be within the realm of common sense. *As such, we are not persuaded that the replacement data must be the same size or configuration, or both, as the decoded data.*

Id. (emphasis added). In this *KSR*-expanded scenario, the host could be "'satisfied' with returned data [that does] not have … the same size … [as] the decoded data," *id.*, despite the specification.

The Board's reliance on *KSR* and obviousness law to essentially fashion new intrinsic evidence is a clear legal error. *KSR* only allows the Board, with appropriate reasoning, to combine multiple prior art teachings to arrive at the claimed invention, 550 U.S. at 420; it does not permit it to expand the scope of the claims themselves. Regardless of whether one of ordinary skill in the art could modify the host to be "satisfied" with an indicator bit, that is a *different* solution to the garbled code problem than the one the '267 patent discloses and claims (see above). As described above, in

the claims and specification of the '267 patent, the amount of replacement "predetermined data" matches the amount of garbled data requested by the host.

Indeed, returning the amount of replacement "predetermined data" that the host originally requested makes perfect sense in the context of an invention that solves the garbled code problem within the flash memory device itself. As discussed above, it allows the flash memory device to work properly with preexisting hosts that operate according to existing industry standards, rather than solely with some new type of host that is specifically programmed to solve the garbled code problem at the host side by converting an indicator bit into the requested data. This new kind of host, moreover, is never described in the '267 patent.

III. The Board's Obviousness Analysis is Wrong on the Law and the Facts

The Board's obviousness analysis of Sharon and Bennett is also flawed, regardless of how the Court resolves the claim construction dispute. The Board concluded that it would be obvious to modify the Sharon system in two steps: first, by adding the indicator bit from Bennett; and second, by transmitting this bit to the host in place of garbled code. A17–18. It was wrong on both counts.

A. The Board Erred by Finding it Obvious to Import the Indicator Bit Into the Sharon System because this Modification Would Make Parts of the Sharon System Superfluous

The first combination step represents an error of law. The Board's factual findings do not lead to the legal conclusion of obviousness because the indicator bit

from Bennett cannot be imported into the Sharon system without rendering part of that system superfluous.

In re NTP, 654 F.3d 1279 (Fed. Cir. 2011) controls here. *NTP* involved two references that disclosed different flavors of the same element. 654 F.3d at 1298-99. One disclosed an "RF network" that did not have an interface element, and the other disclosed a similar "RF network" that did. *Id.* The Board combined them by importing into the first reference's system an *additional* RF network of the second kind—meaning that the combined system had two RF networks. *Id.* This Court rejected this obviousness analysis as a matter of law, in part because "adding an RF network to [the first reference] would render the RF network [already present] superfluous." *Id.*

So too here. There is no dispute that Sharon already discloses an erase-status identifier, called a "flag cell," that does exactly what Bennett's indicator bit would do: namely "indicate an unwritten page." A145 ¶ 78 (describing "allocating one or more flag cells, in the management portion of the page … [which] if found to be in the leftmost state, indicate an unwritten page."). Just as in *NTP*, the Board nevertheless found that it would have been obvious to import a *second* such identifier—Bennett's indicator bit—into the Sharon system in order to meet the limitations of the claims at issue. A16.

It is important to emphasize that the Board's combination uses the indicator bit in addition to, not instead of, the flag cell, *id.* ("The flag cells set in Sharon would

remain set, even under the modification in view of Bennett.") (citations omitted), and indeed *requires* both in order to meet different elements of claim 3. A18 ("[T]he transmitted ES bit and the non-transmitted flag cell are separate elements" of the claim 3 combination). Because the indicator bit does everything that Sharon's flag cell does—namely, identifying unwritten pages—it makes the flag cell superfluous and the combination improper.

The Board did not find that Sharon's flag cell does anything that Bennett's indicator bit does not, A16, and the factual findings it made instead fail to distinguish *NTP*. First, it observed that the indicator bit has a "new use" in the combination: it not only identifies unwritten pages, as the flag bit does, but also goes to the host as "predetermined" or "replacement data." A16. But the issue is not whether the new element (the indicator bit) has any features or uses beyond those of the old one (the flag bit)—if it didn't, there would be no point in introducing it to the combination. In *NTP* itself, for instance, the second RF network had an "interface" feature that the original RF network lacked. *Id.* at 1298. The issue is rather whether the new element makes the old one superfluous. *Id.* at 1298-99. The second indicator bit does that here, just as the second "RF network" did in *NTP*. *Id.*

The same is true of the Board's second finding: that Sharon's flag bit would not "need to be extinguished in order to send the [indicator] bit as replacement data." A16. The issue is not whether the new elements requires removal of the old one—for example, there was no dispute in *NTP* that both RF networks could have operated

simultaneously, *id.* at 1298-99—but rather whether the new element renders the old one meaningless by subsuming all its functions. *Id.* Because there is no dispute that the Bennett's indicator bit subsumes all the flag cell's functions, the Board's obviousness analysis was an error of law even if its factual findings are true. The patchwork of overlapping and unnecessary structure it proposes is legally improper; "[t]his type of piecemeal analysis is precisely the kind of hindsight that the Board must not engage in." *NTP*, 654 F.3d at 1299.

B. Sharon Teaches Away From Sending Bennett's Indicator Bit to the Host

The Board's modification of Sharon in view of Bennett is also factually flawed because Sharon unambiguously teaches away from sending a one-bit response to the host, and indeed lists its ability to send the host just as much data as the host requests as a distinguishing feature of its invention. A142 ¶ 43 ("That the size of the original data is preserved distinguishes the device of the present invention from similar prior art devices that compress data to be stored in their memories."). Accordingly, it would not be obvious to modify Sharon to send a one-bit response. *See U.S. v. Adams*, 383 U.S. 39, 52 (1966) (finding a claim non-obvious because the prior art taught away from the claimed solution); *see also In re Gurley*, 27 F.3d 551, 555 (Fed. Cir. 1994).

The Board applied a clearly unreasonable reading of Sharon. It found that Sharon does not teach away from reducing the size of the data transmitted per se, but only from doing so *by way of data compression*. The Board said:

> The section of Sharon relied upon by Phison is concerned with the data compression and encryption, not with the application of replacement data, as provided for in the claims and in Bennett. We are not persuaded that any apparent proscription against compression would be understood as a prohibition against the use of replacement data by those of ordinary skill in the art. *The convolution of replacement with compression is inappropriate, even though a reduction in the amount of data transmitted could be accomplished by both processes*.

A15 (emphasis ours) (citations omitted).

This simply is not correct. Sharon criticizes data compression *because* it reduces the size of the data; it does not criticize reducing the size of the data only when done using data compression. For example, Sharon defines its invention a few paragraphs before the cited section, saying: "[T]he present invention … include[s] the steps of: [] randomizing original data *while preserving a size of the original data* …"; and later "retrieving … data *substantially identical* to the original data," and "exporting" it to the host. A142 ¶ 39 (emphasis ours). It makes no mention of compression. And Sharon goes on to define the "invention" in this way—to include "preserving a size of the original data" and/or "providing retrieved data substantially identical to the original data," and with no mention of compression—on at least *eight* other occasions in the specification and also in every single claim. A141–142 ¶¶ 31–38 (eight paragraphs, each defining the "invention"); A147–149 (claims); *see also* A142 ¶ 42 ("It is greatly preferred that the retrieved data be strictly identical to the original data ….").

Unquestionably, what the "invention" requires is "preserving [the] size of the data"

and sending the host data that is "substantially identical" to the data it requested, and not anything to do with compression specifically. *Id.*

The statement quoted by the Board—"that the size of the original data is preserved distinguishes the device of the present invention from similar prior art devices that compress data to be stored in their memories"—says the same thing. *Id.* ¶ 43. It plainly states, again, that one of the key features of Sharon is preserving data size, and distinguishes data compression only because it inherently fails to do this. The next sentence confirms that while "[d]ata compression can be construed as a form of at least partial randomization," which is one of Sharon's goals, *id.* ¶ 39 (quoted above), it fails because "data compression, by its very nature, reduces the size of the data being compressed." *Id.* ¶ 43.

In short, Sharon clearly teaches away from changing the size of the data transmitted to the host, with or without data compression. The Board read Sharon's teaching backwards. Its finding is not supported by substantial evidence.

CONCLUSION

For each of the reasons set forth above, the Court should vacate the Board's finding of obviousness.

Dated: July 6, 2015

Respectfully submitted,

/s/ David M. Barkan

David M. Barkan
Fish & Richardson P.C.
500 Arguello Street
Redwood City, CA 94063
(650) 839-5070

Attorneys for Appellant,
PHISON ELECTRONICS CORP.

ADDENDUM

Trials@uspto.gov
571-272-7822

Paper 36
Entered: January 28, 2015

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SILICON MOTION TECHNOLOGY CORP.,
Petitioner,

v.

PHISON ELECTRONICS CORP.,
Patent Owner.

Case IPR2013-00473
Patent 8,176,267 B2

Before KEVIN F. TURNER, JONI Y. CHANG, and MATTHEW R. CLEMENTS, *Administrative Patent Judges.*

TURNER, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
Inter Partes Review
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. INTRODUCTION

Silicon Motion Technology Corp. ("Silicon Motion") filed a Petition ("Pet.," Paper 1) requesting *inter partes* review of claims 1–25 of U.S. Patent No. 8,176,267 B2 ("the '267 Patent"). Patent Owner Phison Electronics Corp. ("Phison") filed a Preliminary Response thereto (Paper 6). On January 28, 2014, we instituted an *inter partes* review of claims 1, 3–7, 9–11, 13–17, 21, 22, 24, and 25 on a single ground of unpatentability alleged in the Petition. Paper 7 ("Dec.").

After institution of trial, Phison filed a Patent Owner Response ("PO Resp.," Paper 13) and Silicon Motion filed a Reply thereto ("Reply," Paper 18). An oral argument was held on September 17, 2014. The transcript of the oral hearing has been entered into the record. Paper 35.

Silicon Motion filed a Motion to Exclude (Paper 25, "Pet. Mot. to Exclude") certain evidence submitted by Phison. Phison filed an Opposition (Paper 29) and Silicon Motion filed a Reply (Paper 33). Phison filed a Motion to Exclude (Paper 23, "PO Mot. to Exclude") the Declaration of Dr. Daniel Foty submitted by Silicon Motion. Silicon Motion filed an Opposition (Paper 31) and Phison filed a Reply (Paper 32). Phison also filed a Motion for Observation (Paper 22, "Obs.") on certain cross-examination testimony of Silicon Motion's Declarant, Dr. Foty, and Silicon Motion filed a Response (Paper 30, "Obs. Resp.").

We have jurisdiction under 35 U.S.C. § 6(c). This Final Written Decision is issued pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73.

Silicon Motion has shown by a preponderance of the evidence that all claims for which trial is instituted, claims 1, 3–7, 9–11, 13–17, 21, 22, 24, and 25 of the ’267 Patent, are unpatentable.

A. Related Matters

Silicon Motion indicates that a complaint alleging infringement of the ’267 Patent was filed November 15, 2012. Pet. 2. *See Phison Electronics Corp. v. PNY Technologies, Inc.*, Civil Action No. 1:12-cv-01478-GMS (D. Del.). PNY Technologies, Inc. is acknowledged as a real party-in-interest in the instant proceeding. *Id*. Another patent, U.S. Patent No. 7,518,879 B2, also assigned to Phison, is also the subject of that litigation, and also the subject of an *inter partes* review, IPR2013-00472, with PNY Technologies, Inc. as its petitioner. *Id.* at 2-3.

B. The ’267 Patent (Ex. 1001)

The subject matter of the ’267 Patent relates to methods of accessing data in a flash memory storage device. Ex. 1001, Abs. The processes prevent a host from reading garbled codes when the system uses a data perturbation module. *Id*. at 1:20–24, 2:7–10. A data perturbation module encodes the data before they are transmitted to the flash memory and decodes the data after they are read from the flash memory to provide security of the data to be protected. *Id*. at 1:47–54.

When the blocks of memory within the flash memory device are initialized, it is not done through a write command, i.e., the data are not encoded by the data perturbation module, such that when data are read from the new blocks, through the data perturbation module, unrecognizable garbled code is produced. *Id*. at 1:55–67. The specification of the '267 Patent provides that when a block to be read is a new block, it replaces the values that would otherwise be read from the requested memory location with predetermined data. *Id*. at 11:59–67.

C. Illustrative Claim

The '267 Patent includes claims 1–25, of which a trial was instituted on claims 1, 3–7, 9–11, 13–17, 21, 22, 24, and 25. Of those, claims 1, 11, and 22 are independent claims. Independent claim 1 is reproduced below:

> 1. A data accessing method, suitable for a flash memory storage device having a data perturbation module, wherein a flash memory of the flash memory storage device has a plurality of physical blocks, and the physical blocks are grouped into at least a data area and a spare area, the data accessing method comprising:
>
> receiving a read command from a host, and obtaining a logical block to be read and a page to be read from the read command;
>
> determining whether a physical block in the data area corresponding to the logical block to be read is a new block;
>
> transmitting a predetermined data to the host when the physical block corresponding to the logical block to be read is the new block; and

> decoding data read from the physical block corresponding to the logical block to be read by the data perturbation and transmitting the decoded data to the host when the physical block corresponding to the logical block to be read is not the new block.

D. Prior Art Relied Upon

The following prior art references were relied upon in the instituted ground of unpatentability:

Bennett	US 2007/0113030	May 17, 2007	Ex. 1003
Sharon	US 2008/0151618	June 26, 2008	Ex. 1002

E. Ground of Unpatentability Instituted for Trial

The following table summarizes the challenge to patentability that was instituted for *inter partes* review:

References	Basis	Claims challenged
Sharon and Bennett	§ 103	1, 3–7, 9–11, 13–17, 21, 22, 24, and 25

II. ANALYSIS

A. Claim Construction

In an *inter partes* review, claim terms in an unexpired patent are given their broadest reasonable construction in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b). Claim terms also are given their ordinary and customary meaning, as would be understood by one of ordinary skill in the art in the context of the entire disclosure. *In re Translogic Tech, Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007).

Predetermined Data

The claim term "predetermined data" is present in all three independent claims. We determined, in the Decision to Institute, that "predetermined data" is different than "decoded data," also used in the claims, and that "predetermined data" would have been understood as "replacement data." Dec. 6.

Claim 3, for example, recites "recording an indicator for each of the physical blocks to indicate that the physical block is the new block during a card activation process performed to the flash memory storage device," with dependent claims 13 and 24 reciting similar limitations. Given such explicit limitations provided in the dependent claims, we concluded that the broadest reasonable construction consistent with the Specification would include that the predetermined data necessarily are "stored before user operation." *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1314–15 (Fed. Cir. 2005) ("[T]he presence of a dependent claim that adds a particular limitation gives rise to a presumption that the limitation in question is not present in the independent claim."). Therefore, we construed "predetermined data" to mean "replacement data," and applied that construction in determining the persuasiveness of the ground of unpatentability discussed below.

Phison accepted this construction for "predetermined data." PO Resp. 15. Silicon Motion requested reconsideration of this claim construction (Paper 10, 3), but we were not persuaded of error in the adopted claim construction. Paper 14, 3.

New Block

In addition, although not contested initially by the parties, we also construed the claim term "new block." Dec. 8. The Specification of the '267 Patent provides that "physical blocks (i.e., new blocks) . . . are just initialized (i.e., data stored therein is 0xFF)." Ex. 1001, 8:17–19. Thus, we construed "new blocks" as physical blocks that have been initialized with data as 0xFF. Dec. 8. We note that Phison disputes this construction, but we continue to adopt it for this proceeding, although we do not determine that the construction is dispositive to the conclusion reached herein. *See* PO Resp. 19, n.2.

B. Principles of Law

To prevail in its challenges to the patentability of the claims, Silicon Motion must prove unpatentability by a preponderance of the evidence. 35 U.S.C. § 316(e); 37 C.F.R. § 42.1(d). A claim is unpatentable under 35 U.S.C. § 103(a) if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). To establish obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *See CFMT, Inc. v. Yieldup Int'l Corp.*, 349 F.3d 1333, 1342 (Fed. Cir. 2003); *In re Royka*, 490 F.2d 981, 985 (CCPA 1974).

A patent claim composed of several elements, however, is not proved

obvious merely by demonstrating that each of its elements was known, independently, in the prior art. *KSR Int'l Co.*, 550 U.S. at 419. In that regard, for an obviousness analysis it is important to identify a reason that would have prompted one of skill in the art to combine prior art elements in the way the claimed invention does. *Id.* However, a precise teaching directed to the specific subject matter of a challenged claim is not necessary to establish obviousness. *Id.* Rather, obviousness must be gauged in view of common sense and the creativity of an ordinarily skilled artisan. *Id.* Moreover, obviousness can be established when the prior art itself would have suggested the claimed subject matter to a person of ordinary skill in the art. *In re Rinehart*, 531 F.2d 1048, 1051 (CCPA 1976).

We analyze the instituted grounds of unpatentability in accordance with the above-stated principles.

C. Claims 1, 3–7, 9–11, 13–17, 21, 22, 24, and 25 – Alleged Obviousness over Sharon and Bennett

Silicon Motion asserts that claims 1–7 and 9–25 of the '267 Patent are unpatentable over Sharon and Bennett under 35 U.S.C. § 103, of which trial was instituted against claims 1, 3–7, 9–11, 13–17, 21, 22, 24, and 25[1]. Dec. 18.

Sharon is directed to a technique for reducing high-block or page-error rates by transforming the user data bits into a pseudo-random bit

[1] We found Silicon Motion to be unpersuasive with respect to claims 2, 12, 18–20 and 23 being obvious over Sharon and Bennett. Dec. 14–15.

sequence that is programmed into the flash memory. Ex. 1002 ¶ 62. Sharon discloses a flash memory device, *id*. ¶ 4, having flash controller 44, which executes flash management software 48 that allows for randomization and derandomization. *Id*. ¶ 27. Sharon also incorporates by reference the disclosure of U.S. Patent No. 5,404,485 to Ban that discusses the use of addressable logical blocks that are written to, and erased in, a flash memory device. *Id*. ¶ 73. Also, in particular, Sharon provides:

> It should be noted that the above method of applying a transformation to the data bits assumes that the flash memory is being programmed. When a flash page is erased all of the cells of the page are set to the left-most state or voltage level (as illustrated in FIGS. 1A and 1B) and all the cells are assumed to contain the fixed all-1's data pattern. This might cause a confusion with a page that was actually programmed to the all-1's bit sequence, but that, according to the present invention, represents some other data bit sequence. However, this can be handled by the application using the flash memory device *being able to distinguish a page that was not written yet from a page that was written*. This is easy to do and is well known in the prior art of flash management systems, for example by allocating one or more flag cells, in the management portion of a page, that are always written as part of the page programming operation, and thus if found to be in the leftmost state, indicate an unwritten page. So a page found to be unwritten is interpreted according to the standard prior art logic, while a page found to be written is interpreted according to the methods of the present invention.

Ex. 1002 ¶ 78 (emphasis added).

Per our discussion in the Decision to Institute, we were not persuaded that Sharon discloses the appropriate logic to send predetermined data in

place of read data for new blocks. Dec. 10. In the context of this ground, Silicon Motion also cites to Bennett for this particular claim limitation. *See, e.g.*, Pet. 29–31.

Bennett is directed to methods of managing erase operations. Ex. 1003, Abs. Figures 3A and 3B of Bennett show the physical and logical grouping of memory cells. Bennett details that some systems utilize a flag on a read operation from an erased block. *Id*. ¶ 153. Such systems "return an 'Erased Status' (e.g., in the MS-PRO Status byte the 'ES' bit) and/or erased data in response to a read operation from an Erased block." *Id*. ¶ 155; *see also id*. ¶ 161. We are persuaded that this process is equivalent to transmitting predetermined data to the host, where the status indicator would be sent in place of data in the block.

Silicon Motion argues that Sharon and Bennett are both directed to methodologies in flash memory systems, Pet. 28, and argues that the standard prior art logic described in Sharon, Ex. 1002 ¶ 78, would have been understood to include Bennett's process of allocating one or more flag cells to indicate the erased states of sectors. Pet. 29–30. We are persuaded that one of ordinary skill in the art would have looked to Bennett in determining the logic applied in Sharon when a determination is made that a block is unwritten. We are persuaded further that the method step of "transmitting a predetermined data to the host when the physical block corresponding to the logical block to be read is the new block," per claim 1 and equivalent elements in claims 11 and 22, would have been obvious in view of the teachings of Sharon and Bennett.

With respect to claims 3–7, 9, 10, 13–17, 21, 24 and 25, Silicon Motion argues that Sharon distinguishes between pages that have been written to and pages that were not written to, and that Bennett discloses the use of an erased flag as an indicator. Pet. 35. Silicon Motion also points to Bennett's discussion of reformatting of memory to reformat as new. *Id.* We are persuaded that performing such a process upon activation of the device would have been obvious, as would providing the indicators in a spare or redundant area during such a process. In addition, Bennett describes the use of a Group Address Table (GAT) to keep track of the mapping between logical groups of sectors and their corresponding megablocks, Ex. 1003 ¶ 61, where a flag may be set therein as "logically" erased. *Id.* ¶ 161.

Phison presents several arguments as to why Sharon and Bennett fail to meet the requirements of the claims and how Silicon Motion has failed to provide an adequate reason to modify the references to reach the claimed invention. PO Resp. 15–34. Silicon Motion responds to these arguments. Reply 1–15. We address each argument in turn below.

Phison begins by arguing that the ES bit of Bennett is not "replacement data" as recited in claim 1. PO Resp. 15. According to Phison, the recited "replacement data" must replace the decoded data that would have been sent to the host, such that "the host thinks that it received the data that it requested." PO Resp. 16. Phison continues that "[r]eplacement data must be able to take the place of the requested data," with the host having an expectation that it will receive a page and/or sector of data. PO Resp. 17–18. Silicon Motion counters that the instant claims

are not so limited, and that the host need not specify an amount of data to read. Reply 3. Silicon Motion cites to Dr. Foty's testimony that the command to read multiple blocks in the SD Card Specification (Ex. 1004) does not require an amount of data to be specified. Reply 5; Ex. 1013 ¶¶ 29–30. Silicon Motion also argues that claim 1 does not require an entire page of data to be read, nor does the claim provide for "an amount of the requested data" as a limitation. Reply 5–6. We find Silicon Motion's arguments to be persuasive.

We continue to be persuaded that "predetermined data" would have been understood as "replacement data." Dec. 6. We find nothing in claim 1, or the other independent claims, that would require that the "replacement data" must be the same or similar to the decoded data such that the host thinks that it received the requested data. Although the "replacement data" must necessarily replace the decoded data, that does not necessarily mean that it would need to have the same size or structure, or both, of the decoded data. From a computer science perspective, there are multiple ways a host could be "satisfied" with returned data and not have that data be the same size. It is important to remember that "in many cases a person of ordinary skill will be able to fit the teachings of multiple patents together like pieces of a puzzle," *KSR Int'l Co.*, 550 U.S. at 420, such that curing obvious problems would be within the realm of common sense. As such, we are not persuaded that the replacement data must be the same size or configuration, or both, as the decoded data.

Phison also argues that the ES bit of Bennett is an "indicator," which is separate from "replacement data" in the context of the '267 Patent. PO Resp. 18–19. Phison continues that "different claim terms are legally presumed to have different meanings unless the specification of prosecution history dictates otherwise." PO Resp. 19–20. Phison argues that the alignment of the ES bit with the new block indicator in the specification of the '267 Patent demonstrates that the ES bit cannot also qualify as the "predetermined data." PO Resp. 20–21. Silicon Motion counters that the ES bit can be replacement data because Bennett details that the flash memory can return the ES bit, the erased data, or both, such that the ES bit can act as replacement data. Reply 11–12. We agree with Silicon Motion.

Although Phison is correct that the ES bit of Bennett can be an indicator, it can also serve as replacement data. As detailed in Bennett: typical performance requirements include an "[a]bility to return an 'Erased Status' (e.g., in the MS-PRO Status byte the 'ES' bit) and/or erased data in response to a read operation from an Erased block." Ex. 1003 ¶¶ 154, 155. That the ES bit of Bennett might closely resemble the new block indicator in the '267 Patent does not mean that the ES bit cannot act as replacement data. Phison is correct that the ES bit cannot satisfy both of the separately recited claim terms, as discussed below, but given the disclosure of Bennett, it is clear that the ES bit can act as replacement data.

Phison disputes that Silicon Motion provided an adequate rationale to combine Sharon and Bennett for several reasons. PO Resp. 21–33. Phison argues that modifying Sharon to transmit the ES bit would create an

inoperable system. PO Resp. 21–25. Phison argues that, if the requested bytes of data are not returned in the system of Sharon, an error will be generated. PO Resp. 22–24. Phison further relies on the testimony of its Declarant , Dr. Andrew Wolfe, to argue that Sharon and Bennett would result in an inoperable system. Ex. 2005 ¶¶ 39–46, 55–58. Although we credit Dr. Wolfe's testimony, we do not find Phison's arguments to be persuasive.

Dr. Wolfe discusses the functioning of the Windows operating system in performing a read from storage, with his "understanding that other host operating systems have similar operating principles when operating as hosts for flash memory storage devices." Ex. 2005 ¶ 44. And althoughDr. Wolfe continues that "one of ordinary skill in the art would understand that a host computer requires the receipt of the amount of data that it requested from a storage device," (*id*. ¶ 45), we are persuaded that this finding is undercut by the disclosure of Bennett, discussed above, where the ES bit or erased data are returned in response to a read operation from an Erased block. Although Dr. Wolfe considered the modification of Sharon, the process of returning a single bit in Bennett must be considered in determining the obviousness of any modification. Dr. Wolfe clearly considered Bennett—which also is directed to flash memory—but does not appear to have considered the combined teachings of Sharon and Bennett" (*see id.* ¶¶ 59–63). As discussed by Silicon Motion, Dr. Wolfe has acknowledged that his analysis was limited to the Windows ReadFile function, that other specific specifications were not considered, and that no specific configuration of the

host system in Sharon is specified. Reply 7–8; Ex. 1012, 87:3–15, 128:5–129:19, 131:8–23. Therefore, although we may agree with Dr. Wolfe that Sharon operating according to the Windows ReadFile function may return an error when a single bit is received, claim 1 and Sharon are not so limited. As such, we are not persuaded that the modification of Sharon by the processes in Bennett would have rendered the system of Sharon to be inoperable.

Phison also argues that Sharon requires the transmission of the requested data and not a status bit, with Sharon teaching away from the proposed modification. PO Resp. 25–28. Phison argues that Sharon discourages ordinarily skilled artisans from compressing or reducing the amount of data returned to the host, and that the return of the precise data stored is disclosed by Sharon to be a fundamental feature thereof. PO Resp. 27–28. We do not agree.

The section of Sharon relied upon by Phison (Ex. 1002 ¶ 43) is concerned with the data compression and encryption, not with the application of replacement data, as provided for in the claims and in Bennett. We are not persuaded that any apparent proscription against compression would be understood as a prohibition against the use of replacement data by those of ordinary skill in the art. The convolution of replacement with compression is inappropriate, even though a reduction in the amount of data transmitted could be accomplished by both processes. As such, we do not find it availing that "there is no disclosure with the Sharon reference to indicate that the requirement to return the requested data is changed for

unwritten pages" (PO Resp. 28) because that motivation is found within Bennett, as discussed above.

Phison also argues that there would have been no reason to add the use of the ES bit to Sharon because Sharon already includes erase status bits. PO Resp. 29. That formulation, however, does not take into account the new *use* of the bits as motivated by Bennett. The flag cells set in Sharon (Ex. 2005 ¶ 78) would remain set, even under the modification in view of Bennett. Neither Sharon nor Bennett suggests that the flag cell set in Sharon to indicate an unwritten page would need to be extinguished in order to send the ES bit as replacement data. Phison also cites to *In re NTP*, 654 F.3d 1279, 1298 (Fed. Cir. 2011), for its discussion of the addition of elements from one reference into another, thereby making the first elements superfluous. PO Resp. 29. However, the citation is inapt because the combination of Sharon and Bennett would not render the flags cells of Sharon or the ES bit of Bennett superfluous.

Phison also notes that Sharon purposely does not return its erased block indicator, strongly indicating that such a modification would not have been motivated. PO Resp. 30–31. However, absence of evidence is not evidence of absence. The fact that Sharon does not transmit flags cells means that any potential finding of Sharon to be anticipatory over the claims would have been in error, but does not constitute a negative teaching or a teaching away from the functionalities of Bennett. Even if Sharon does not disclose the exact methods of the challenged claims, it does not impact whether the processes of Sharon can be modified according to Bennett.

Based on the above discussion, we are persuaded that Sharon and Bennett render the challenged claims obvious.

Phison also alleges that an insufficient rationale to modify Sharon in view of Bennett has been proffered by Silicon Motion, and accepted by the Board, and that such a combination is only motivated by improper hindsight reconstruction. PO Resp. 31–33. We determined previously that the processes in Bennett provide context to the standard prior art logic described in Sharon and provide illustrations of how the situation of new blocks in Sharon could have been handled. Dec. 12. We did not need to resort to improper hindsight reconstruction because Bennett discloses how the reading of new blocks can be handled, adding to the information in Sharon, that addresses the "confusion" problem by integrating a randomizer into the flash memory system. Pet. 29. By providing additional disclosure, Bennett suggests how certain reading processes from memory should be handled, without requiring the disclosure of the '267 Patent to serve as a roadmap.

In addition, Phison argues that the ES bit of Bennett cannot be used to teach or suggest both the claimed "replacement data," in claim 1, and "an indicator," in dependent claim 3, because they are separately claimed elements. PO Resp. 33–34. We agree with the principle outlined by Phison, but we are not convinced that the same feature was found to be equivalent to two distinct aspects of the claims.

We have discussed, and Silicon motion has cited (Pet. 36), both the flag cells used to distinguish between pages that have been written to or not, and the ES bit. We are not persuaded that the ES bit, sent as replacement

data, is the same as the flag cell, which is not transmitted back to the host. As discussed above, the disclosure of the setting of the flag cell does not preclude its alternative uses as disclosed in Bennett. In effect, once the bit is sent as replacement data, the flag in Sharon remains set to the selected value and still serves the same purpose disclosed. Claim 3 only requires "recording an indicator," which is then used in making a determination as to whether the logical block is a new block. In other words, we are persuaded that the transmitted ES bit and the non-transmitted flag cell are separate elements that can render obvious elements of claims 1 and 3.

We are persuaded that Silicon Motion has shown by a preponderance of the evidence that claims 1, 3–7, 9–11, 13–17, 21, 22, 24, and 25 of the '267 Patent are unpatentable over Sharon and Bennett under 35 U.S.C. § 103.

D. Motions to Exclude

The party moving to exclude evidence bears the burden of proof to establish that it is entitled to the relief requested—namely, that the material sought to be excluded is inadmissible under the Federal Rules of Evidence. *See* 37 C.F.R. §§ 42.20(c), 42.62(a).

Phison seeks to exclude Dr. Foty's Declaration (Ex. 1013) on the grounds that Dr. Foty lacks sufficient expertise in the subject matter of this proceeding to qualify as one of ordinary skill in the art or an expert. PO Mot. to Exclude 1. Alternatively, Phison seeks to exclude paragraphs 20–24 and 26 of the same Declaration because they are based on the wrong

standard for claim construction. *Id*. We are not persuaded that Dr. Foty's Declaration, or portions thereof, should be excluded.

Phison argues that Dr. Foty has little or no experience with flash memory systems, where the undisputed level of ordinary skill in the art is "a Bachelor's degree in Electrical Engineering and between 1–2 years of experience in the design and/or analysis of flash memory storage systems or an equivalent combination of education and experience." PO Mot. to Exclude 2 (citing Ex. 2005 ¶20, emphasis omitted). Based on Dr. Foty's experience, Phison argues that Dr. Foty "would not even qualify as a person of ordinary skill in the art of the '267 patent." PO Mot. to Exclude 3. We do not agree.

Phison's analysis focuses on the "design" of such systems but not the "analysis" of such systems. Although Dr. Foty may not have the *design* experience with flash memory storage systems, given his testimony, we cannot say that he has no experience with such systems. Phison has not explained sufficiently why flash memory systems is so specialized a field that Dr. Foty, having experience with multiple types of memory systems, would not be qualified to offer opinions in this proceeding. We are persuaded that Dr. Foty has sufficient experience to act as an expert in the instant proceeding, and do not find sufficient reason to exclude his testimony.

Phison also argues that "Dr. Foty ignored the Board's claim construction and based his opinion on a fundamentally flawed claim construction methodology," with respect to the claim term "predetermined

data," such that portions of his testimony, paragraphs 20–24 and 26 should be excluded. PO Mot. to Exclude 3. Although it may have been preferable for Dr. Foty to apply our construction of the term "predetermined data" instead of a dictionary definition, this goes to the weight given to his testimony but does not require that his testimony be excluded. Dr. Foty's view of the claim term, as one of ordinary skill in the art, is still relevant and illustrates that our construction is consistent that the ordinary use of that term. As such, we are not persuaded that paragraphs 20–24 and 26 of Dr. Foty's Declaration (Ex. 1013) should be excluded.

Silicon Motion moves to exclude specific documents as evidence (Paper 25), but subsequently withdrew motions to exclude with respect to most of the documents it sought to exclude. Paper 33. Silicon Motion argues that Exhibit 2005, Declaration of Dr. Wolfe, is deficient under the Federal Rules of Evidence. *Id*. at 3–5. Silicon Motion maintains that Dr. Wolfe's statement about his Declaration being made under penalty of perjury (Ex. 2005, 16) does not meet the requirements of 37 C.F.R. § 42.2. Paper 33, 4–5.

We need not reach the merits of Silicon Motion's Motion to Exclude because, as explained above, even if the disputed evidence is considered, we have concluded that Silicon Motion has demonstrated, by a preponderance of the evidence, that the challenged claims are unpatentable. Accordingly, Silicon Motion's Motion to Exclude is dismissed as moot.

F. Motion for Observation

Phison's observations are directed to the cross-examination testimony of Dr. Foty (Ex. 2009), who was deposed after Silicon Motion filed its Reply. We have considered Phison's observations and Silicon Motion's responses in rendering our decision, and have accorded the testimony the appropriate weight. *See* Obs. 1–3; Obs. Resp. 1–4. Although Phison's observations about the functionalities of Sharon and Bennett have been considered, they do not change our conclusions about the obviousness of the subject claims in view of Sharon and Bennett, as discussed above.

III. CONCLUSION

We conclude Silicon Motion has shown by a preponderance of the evidence that claims 1, 3–7, 9–11, 13–17, 21, 22, 24, and 25 of the '267 Patent are unpatentable under 35 U.S.C. § 103(a) as obvious over Sharon and Bennett.

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that claims 1, 3–7, 9–11, 13–17, 21, 22, 24, and 25 of the '267 Patent are held unpatentable;

FURTHER ORDERED that Phison's Motion to Exclude is *denied*;

FURTHER ORDERED that Silicon Motion's Motion to Exclude is *dismissed*; and

FURTHER ORDERED that because this is a Final Written Decision, parties to the proceeding seeking judicial review of the Decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

IPR2013-00473
Patent 8,176,267 B2

For PETITIONER:

Melvin D. Chan
Gary T. Aka
AKA CHAN LLP
mel@akachanlaw.com
mail@akachanlaw.com

For PATENT OWNER:

Joshua A. Griswold
David M. Hoffman
FISH & RICHARDSON P.C.
griswold@fr.com
hoffman@fr.com
IPR23490-0009IP1@fr.com

Trials@uspto.gov
571-272-7822

Paper 7
Entered: January 28, 2014

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SILICON MOTION TECHNOLOGY CORP.
Petitioner

v.

PHISON ELECTRONICS CORP.
Patent Owner

Case IPR2013-00473
Patent 8,176,267

Before KEVIN F. TURNER, JONI Y. CHANG, and MATTHEW R. CLEMENTS, *Administrative Patent Judges.*

TURNER, *Administrative Patent Judge*.

DECISION
Institution of *Inter Partes* Review
37 C.F.R. § 42.108

I. INTRODUCTION

Silicon Motion Technology Corp. ("Silicon Motion") filed a Petition ("Pet.," Paper 1) requesting *inter partes* review of claims 1-25 of U.S. Patent No. 8,176,267 ("the '267 Patent"). Patent Owner Phison Electronics Corp. ("Phison") filed a Preliminary Response thereto ("Prelim. Resp.," Paper 6). We have jurisdiction under 35 U.S.C. § 314.

The standard for instituting an *inter partes* review is set forth in 35 U.S.C. § 314(a), which provides:

> THRESHOLD—The Director may not authorize an inter partes review to be instituted unless the Director determines that the information presented in the petition filed under section 311 and any response filed under section 313 shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.

We are persuaded that the information presented in the Petition demonstrates that there is a reasonable likelihood that Silicon Motion will prevail in challenging claims 1, 3-7, 9-11, 13-17, 21, 22, 24, and 25 as unpatentable under 35 U.S.C. § 103. Pursuant to 35 U.S.C. § 314, we hereby authorize an *inter partes* review to be instituted as to claims 1, 3-7, 9-11, 13-17, 21, 22, 24, and 25 of the '267 Patent.

A. Related Matters

Silicon Motion indicates that a complaint alleging infringement of the '267 Patent was filed November 15, 2012. Pet. 2. *See Phison Electronics Corp. v. PNY Technologies, Inc.*, Civil Action No. 1:12-cv-01478-GMS (D.

Del.). PNY Technologies, Inc. is acknowledged as a real party-in-interest in the instant proceeding. *Id*. Another patent, U.S. Patent No. 7,518,879, also assigned to Phison, is also the subject of that litigation, and also the subject of *inter partes* review, IPR2013-00472, with PNY Technologies, Inc. as its petitioner. *Id.* at 2-3.

B. The Invention of the '267 Patent (Ex. 1001)

The invention of the '267 Patent relates to methods of accessing data in a flash memory storage device. Ex. 1001, Abs. The processes prevent a host from reading garbled codes when the system uses a data perturbation module. *Id*. at 1:20-24, 2:7-10. A data perturbation module encodes the data before they are transmitted to the flash memory and decodes the data after they are read from the flash memory to provide security of the data to be protected. *Id*. at 1:47-54.

When the blocks of memory within the flash memory device are initialized, it is not done through a write command, i.e., without encoding, such that when data are read from the new blocks, through the data perturbation module, expecting encoded data, unrecognizable garbled code are produced. *Id*. at 1:55-67. The specification of the '267 Patent provides that when a block to be read is a new block, it replaces the values that would otherwise be read from the requested memory location with predetermined data. *Id*. at 11:59-67.

C. Challenged Claims

Silicon Motion challenges independent claims 1, 11, and 22, as well as dependent claims 2-10, 12-21 and 23-25, in its Petition. Claim 1 is reproduced below:

> 1. A data accessing method, suitable for a flash memory storage device having a data perturbation module, wherein a flash memory of the flash memory storage device has a plurality of physical blocks, and the physical blocks are grouped into at least a data area and a spare area, the data accessing method comprising:
>
> receiving a read command from a host, and obtaining a logical block to be read and a page to be read from the read command;
>
> determining whether a physical block in the data area corresponding to the logical block to be read is a new block;
>
> transmitting a predetermined data to the host when the physical block corresponding to the logical block to be read is the new block; and
>
> decoding data read from the physical block corresponding to the logical block to be read by the data perturbation and transmitting the decoded data to the host when the physical block corresponding to the logical block to be read is not the new block.

D. Prior Art Relied Upon

Silicon Motion relies upon the following prior art references in its alleged grounds of unpatentability:

Asnaashari	US 5,928,370	Jul. 27, 1999	Ex. 1007
Morley	US 6,549,446	Apr. 15, 2003	Ex. 1010

Bennett	US 2007/0113030	May 17, 2007	Ex. 1003
Sharon	US 2008/0151618	Jun. 26, 2008	Ex. 1002
Min	US 2008/0263369	Oct. 23, 2008	Ex. 1009

E. Alleged Grounds of Unpatentability

Silicon Motion seeks to have canceled claims 1-25 of the '267 Patent based on the following alleged grounds of unpatentability:

Reference(s)	**Basis**	**Claims challenged**
Sharon	§ 102	1, 2, 11, 12, 22, and 23
Sharon and Bennett	§ 103	1-7 and 9-25
Sharon and Asnaashari	§ 103	8
Min	§ 102	1-3, 6, 8, and 22-24
Min and Morley	§ 103	2, 6, 11-13, 16, 21, 23, and 24

II. ANALYSIS

A. Claim Construction

As a first step in our analysis for determining whether to institute a trial, we determine the meaning of the claims. In an *inter partes* review, claim terms in an unexpired patent are given their broadest reasonable construction in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b). Under the broadest reasonable construction standard, claims are to be given their broadest reasonable interpretation consistent with the specification, and the claim language should be read in light of the specification as it would be interpreted by one of ordinary skill in the art. *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004).

Silicon Motion argues that all claims should be interpreted according to their plain and ordinary meanings. Pet. 5. The sole "disputed" claim term is that of "predetermined data," where that claim limitation is present in all three independent claims. Silicon Motion does not construe specifically this limitation, but does allege that its use in the claims does not raise a point of novelty at the time of the invention. Pet. 12-13. Phison, on the other hand, has provided a specific construction of the subject limitation. Prelim. Resp. 11-16. We construe the limitation below.

Phison argues that "predetermined data" would have been understood to mean "replacement data stored before user operation," Prelim. Resp. 12, also arguing that "predetermined data" must be different from the also-claimed "decoded data," *id*., and would be consistent with examples of use from the Specification of the '267 Patent. *Id*. at 13-16. We are persuaded of the difference in the claim terms and that the "predetermined data" would have been understood as "replacement data." We are not persuaded, however, that such replacement data would have been stored before user operation necessarily, i.e., that it would have a temporal dependence. The Specification discusses the writing of predetermined data during the card activation process, but we do not conclude that claim 1, and the other independent claims, are so limited.

Claim 3, for example, recites "recording an indicator for each of the physical blocks to indicate that the physical block is the new block during a card activation process performed to the flash memory storage device," with dependent claims 13 and 24 reciting similar limitations. Given such explicit

limitations provided in the dependent claims, we cannot conclude that the broadest, reasonable construction consistent with the Specification would include that the predetermined data are "stored before user operation" necessarily. *See Phillips v. AWH Corp*., 415 F.3d 1303, 1314-15 (Fed. Cir. 2005) ("[T]he presence of a dependent claim that adds a particular limitation gives rise to a presumption that the limitation in question is not present in the independent claim."). Therefore, we construe "predetermined data" to mean "replacement data," and apply that construction in determining the persuasiveness of the grounds of unpatentability discussed below.

We also note that Phison has argued that the Petition improperly lacks proposed claim constructions, thus failing to meet one of the requirements of 37 C.FR. § 42.104(b)(4). Prelim. Resp. 10-11. Phison also quotes from *Norman Noble v. Nutech Ventures*, IPR2013-00101, Paper 14 at 6 (PTAB Jun. 20, 2013), arguing that Silicon Motion's lack of a claim construction should be fatal to the Petition. *Id*. We do not agree. *Norman Noble* was concerned with claim terms whose meanings were not apparent immediately from the intrinsic record. In the instant proceeding, Phison has not alleged that the meanings of any claim terms are not apparent readily, such that adequate evidence to support the construction would be necessary. This does not remove, however, the importance of petitioners supplying a properly supported claim construction when necessary, per 37 C.FR. § 42.104(b)(4).

On the facts of this proceeding, the Petition contains a statement that the claims terms should be provided with the broadest reasonable

construction in light of the specification, Pet. 5, and we do not find that more is required. Silicon Motion may gain no benefit that it might have had from a more specific construction, but the grounds of the Petition will be considered based on the ordinary and customary meaning. Thus, we are not persuaded that the Petition should be denied on such a basis.

In addition, although not contested by the parties, we also construe the claim term "new block." The Specification of the '267 Patent provides that "physical blocks (i.e., new blocks) . . . are just initialized (i.e., data stored therein is 0xFF)." Ex. 1001 at 8:17-19. Thus, we construe "new blocks" as physical blocks that have been initialized with data as 0xFF.

B. Grounds of Unpatentability

i) Anticipation by Sharon

Silicon Motion asserts that claims 1, 2, 11, 12, 22, and 23 of the '267 Patent are anticipated by Sharon under 35 U.S.C. § 102(a) or (e). Sharon is directed to a technique for reducing high block or page error rates by transforming the user data bits into a pseudo-random bit sequence that is programmed into the flash memory. Ex. 1002 ¶ 0062.

Sharon is directed to a flash memory device, *id*. at ¶ 0004, having flash controller 44, which executes flash management software 48 that allows for randomization and derandomization. *Id*. at ¶ 0027. Sharon also incorporates by reference the disclosure of U.S. Patent No. 5,404,485 to Ban that discusses the use of addressable logical blocks that are written to, and

erased in, a flash memory device. *Id*. at ¶ 0073. Also, in particular, Sharon provides:

> It should be noted that the above method of applying a transformation to the data bits assumes that the flash memory is being programmed. When a flash page is erased all of the cells of the page are set to the left-most state or voltage level (as illustrated in FIGS. 1A and 1B) and all the cells are assumed to contain the fixed all-1's data pattern. This might cause a confusion with a page that was actually programmed to the all-1's bit sequence, but that, according to the present invention, represents some other data bit sequence. However, this can be handled by the application using the flash memory device *being able to distinguish a page that was not written yet from a page that was written*. This is easy to do and is well known in the prior art of flash management systems, for example by allocating one or more flag cells, in the management portion of a page, that are always written as part of the page programming operation, and thus if found to be in the leftmost state, indicate an unwritten page. So a page found to be unwritten is interpreted according to the standard prior art logic, while a page found to be written is interpreted according to the methods of the present invention.

Ex. 1002 ¶ 0078 (emphasis added).

Silicon Motion argues that the above-addressed "all-1's data pattern" is equivalent to an FF state, and "is a 'predetermined value' that is stored in the unwritten cells after initialization" and transmitted to the host. Pet. 20. Silicon Motion argues that the process disclosed in Sharon is equivalent to the method step of "transmitting a predetermined data to the host when the physical block corresponding to the logical block to be read is the new

block," as recited in claim 1, with equivalent recitations in the other dependent claims. Pet. 19-21, 24, 26, 27. We are not persuaded.

As Phison points out, Sharon only discloses the use of "standard prior art logic," which is not specific, and suggests that the actual page of non-randomized data is read and transmitted, as opposed to predetermined data, as provided for in the independent claims. Prelim. Resp. 18-19. Thus, although it is possible that the "standard prior art logic" could supply replacement data for new blocks being read, we are not persuaded that this is anything beyond mere speculation. Additionally, Silicon Motion's discussion of the FF state being a predetermined value suggests that Silicon Motion considers the all-1's data pattern to be read out and transmitted in Sharon to satisfy the claim limitation. Pet. 20, 24. However, further to our construction of "predetermined data," we are not persuaded that the reading and transmitting of data found in the specific block is equivalent to the provision of replacement data. In other words, there would be no replacement, only what was there to begin with and decoded. Thus, we are not persuaded that Sharon teaches "transmitting a predetermined data to the host when the physical block corresponding to the logical block to be read is the new block," as recited in claim 1, with equivalent recitations in the other dependent claims.

Therefore, we are not persuaded that Silicon Motion has shown a reasonable likelihood that it will prevail in challenging claims 1, 2, 11, 12, 22, and 23 as anticipated under 35 U.S.C. § 102 by Sharon.

ii) Obviousness over Sharon and Bennett

Silicon Motion asserts that claims 1-7 and 9-25 of the '267 Patent are unpatentable over Sharon and Bennett under 35 U.S.C. § 103. Per our prior discussion, above, we are persuaded that Sharon discloses a system suitable for a flash memory that provides randomization and derandomization of data, where read commands from a host cause a controller to read from appropriate logical blocks. Sharon also details that a determination is made whether a block is unwritten or not, and appropriate logic is applied in each case. Ex. 1002 ¶ 0078. As discussed above, we are not persuaded that Sharon discloses that the appropriate logic would be to send predetermined data in place of read data for new blocks. With respect to that claimed limitation, Silicon Motion also cites Bennett. Pet. 28-31.

Bennett is directed to methods of managing erase operations. Ex. 1003, Abs. Figures 3A and 3B of Bennett show the physical and logical grouping of memory cells. Bennett details that some systems utilize a flag on a read operation from an erased block. *Id*. at ¶ 0153. Such systems "return an 'Erased Status' (e.g., in the MS-PRO Status byte the 'ES' bit) and/or erased data in response to a read operation from an Erased block." *Id*. at ¶ 0155; *see also id*. at ¶ 0161. We are persuaded that this process is equivalent to transmitting predetermined data to the host, where the status indicator would be sent in place of data in the block.

Silicon Motion argues that Sharon and Bennett are both directed to methodologies in flash memory systems, Pet. 28, and argues that the standard prior art logic described in Sharon, Ex. 1002 ¶ 0078, would have

been understood to include Bennett's process of allocating one or more flag cells to indicate the erased states of sectors. Pet. 29-30. We are persuaded that one of ordinary skill in the art would have looked to Bennett in determining the logic applied in Sharon when a determination is made that a block is unwritten. We are persuaded further that the method step of "transmitting a predetermined data to the host when the physical block corresponding to the logical block to be read is the new block," per claim 1 and equivalent elements in claims 11 and 22, would have been obvious in view of the teachings of Sharon and Bennett.

Phison argues that "the Petitioner does not provide a legitimate reason to combine the teachings of Sharon and Bennett," Prelim. Resp. 31, but we are not persuaded that this is correct. As discussed by Silicon Motion, Pet. 28-31, the processes in Bennett provide context to the standard prior art logic described in Sharon and provide illustrations of how the situation of new blocks in Sharon could have been handled. We are persuaded that this is a sufficient rationale to combine the teachings of Sharon and Bennett.

Phison argues that Bennett fails to cure the deficiencies of Sharon. Prelim. Resp. 26. Phison details that the two references address disparate issues, with Sharon directed to methods of reducing error rates and Bennett directed to an improved method of managing erase operations. Prelim. Resp. 26-27. We do not agree. As discussed above, both references are concerned with methodologies in flash memory systems, and it would have been logical for one of ordinary skill in the art to have looked to both in

addressing issues with such devices. *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 420-421 (2007).

Phison continues that Bennett never describes the ES bit/flag as being transmitted "to the host," as recited in the claims. Prelim. Resp. 28. But it need not, as transmission to the host is provided for in Sharon, and both references applied in an obviousness determination need not disclose all of the elements of a claim. Phison also argues that the ES flag is not replacement data because that flag is asserted if the host reads a block, whereas the instant claims require "transmitting a predetermined data," rather than reading erased data. *Id.* However, nothing in the independent claims precludes a read of an erased or a new sector from occurring, only reciting that "predetermined data" be sent back to the host on such an occurrence. As discussed above, we are persuaded that it would have been obvious to have returned the ES bit upon an attempted read of an erased or new block.

In addition, Phison argues that Bennett is concerned with erased blocks and the Petition provides no evidence that Bennett's processes are used to create "new blocks." Prelim. Resp. 29. We do not agree. The Petition provides that "[e]rased blocks have FFs as their content and an erased status." Pet. 29-30. From a compositional standpoint, we discern no physical difference between an erased block that has FFs as its contents and a new block, that has never been written to, with data as 0xFF as its contents, per the above-discussed claim construction. In addition, this distinction is made by Sharon, as discussed above, such that the combination

of Sharon and Bennett also would appreciate any distinction between new blocks and erased blocks.

Phison also argues that this ground of unpatentability, obviousness over Sharon and Bennett, should be rejected as redundant. Prelim. Resp. 21-25. Phison argues that the Petition "makes no effort to distinguish between [the grounds] or otherwise establish why each is necessary." Prelim. Resp. 21. We point out, however, that non-redundancy is not a requirement set forth in the rules for *inter partes* review. Although Phison cites 37 C.F.R. § 42.1(b), the application of that rule "to secure the just, speedy, and inexpensive resolution of every proceeding" is at the Board's discretion. It is not incumbent on the Board to discard grounds found to be redundant if the overall proceeding can be secured in a just, speedy, and inexpensive manner. While it is true that petitioners should explain the relative strengths and weaknesses of their challenges to avoid finding grounds redundant, it is not necessarily a defect in the petition if they do not. In the instant case, we find the ground over Sharon and Bennett more persuasive than the ground over Sharon alone, so that we do not find the grounds to be redundant.

With respect to claims 2, 12, 18-20 and 23, Silicon Motion argues that Bennett discloses that the expected erased blocks shall have all FFs as their content, and when a sector appears to be erased, the system has the ability to return an "Erased Status." Pet. 34, 40, 42-43, and 45. However, if the data returned from the read operation are all FFs, as Silicon motion alleges, then these would be read data and not replacement data. Alternatively, if the ES bit is set to "1" and returned, as discussed with respect to claim 1, the

predetermined data would not be 0x00 or 0xFF, as required by claim 2. Accordingly, we are not persuaded that Silicon Motion has shown that claim 2 would have been obvious over Sharon and Bennett. Additionally, with respect to this ground of unpatentability, Silicon Motion's analyses of claims 12, 18-20 and 23 depend on its analysis of claim 2. Pet. 40, 42-43, 45. Thus, we also are not persuaded as to the obviousness of those claims.

With respect to claims 3-7, 9, 10, 13-17, 21, 24 and 25, Silicon Motion argues that Sharon distinguishes between pages that have been written to and pages that were not written to, and that Bennett discloses the use of an erased flag as an indicator. Pet. 35. Silicon Motion also points to Bennett's discussion of reformatting of memory to reformat as new. *Id*. We are persuaded that performing such a process upon activation of the device would have been obvious, as would providing the indicators in a spare or redundant area during such a process. In addition, Bennett describes the use of a Group Address Table (GAT) to keep track of the mapping between logical groups of sectors and their corresponding megablocks, Ex. 1003 ¶ 0061, where a flag may be set therein as "logically" erased. *Id*. at ¶ 0161.

We are persuaded that Silicon Motion has demonstrated that it has a reasonable likelihood of prevailing in showing that claims 1, 3-7, 9-11, 13-17, 21, 22, 24, and 25 of the '267 Patent are unpatentable over Sharon and Bennett under 35 U.S.C. § 103.

iii) Obviousness over Sharon and Asnaashari

Silicon Motion asserts that claim 8 would have been obvious over the combination of Sharon and Asnaashari. Pet. 46. Silicon Motion argues this ground of unpatentability based on the anticipation ground of Sharon alone. *Id*. However, given that we are not persuaded that Silicon Motion has shown a reasonable likelihood that it will prevail in challenging claim 1 as anticipated under 35 U.S.C. § 102 by Sharon, as discussed above, with claim 8 dependent from claim 1, we likewise are not persuaded by the arguments directed to Sharon and Asnaashari.

iv) Anticipation by Min

Silicon Motion asserts that claims 1-3, 6, 8, and 22-24 of the '267 Patent are anticipated by Min under 35 U.S.C. § 102. Silicon Motion asserts that Min discloses "transmitting a predetermined data to the host when the physical block corresponding to the logical block to be read is the new block," per claim 1. Pet. 49. Phison argues that, like Sharon, Min "merely teaches the prior art technique of reading non-encoded data from erased pages and transmitting to the host" whatever data happen to be stored in the subject erased page. Prelim. Resp. 34-35. We are persuaded by Phison's argument.

The process of Min is illustrated in Figure 4 of Min. In that flowchart, an evaluation is made of whether the data are encrypted (step 405), and the data in the subject page are read out, with or without encryption. Ex. 1009 ¶¶ 0056-0061. As Phison points out, "[t]here is no provision for any type of

predetermined replacement data." Prelim. Resp. 37. Similar to the discussion above of Sharon alone, we are persuaded that Min does not teach or suggest "transmitting a predetermined data to the host when the physical block corresponding to the logical block to be read is the new block," per claim 1, with an equivalent limitation in independent claim 22.

Accordingly, we are not persuaded that Silicon Motion has shown a reasonable likelihood that it will prevail in challenging claims 1-3, 6, 8, and 22-24 as anticipated under 35 U.S.C. § 102 by Min.

v) Obviousness over Min and Morley

Silicon Motion asserts that claims 2, 6, 11-13, 16, and 21, 23, and 24 would have been obvious over the combination of Min and Morley. Pet. 54-59. Silicon Motion argues this ground of unpatentability based on the anticipation ground of Min. Pet. 54. However, given that we are not persuaded that Silicon Motion has shown a reasonable likelihood that it will prevail in challenging claims as anticipated under 35 U.S.C. § 102 by Min, we likewise are not persuaded by the arguments directed to Min and Morley.

III. CONCLUSION

For the foregoing reasons, we determine that the information presented in Silicon Motion's Petition shows that there is a reasonable likelihood that it would prevail with respect to claims 1, 3-7, 9-11, 13-17, 21, 22, 24, and 25 of the '267 Patent. Accordingly, the Petition is granted on the grounds discussed above.

IV. ORDER

It is ORDERED that pursuant to 35 U.S.C. § 314(a), an *inter partes* review is hereby instituted as to claims 1, 3-7, 9-11, 13-17, 21, 22, 24, and 25 of the '267 Patent for the following ground of unpatentability:

Claims 1, 3-7, 9-11, 13-17, 21, 22, 24, and 25 as unpatentable under 35 U.S.C. § 103 over Sharon and Bennett.

It is FURTHER ORDERED that an *inter partes* review is not instituted with respect to any of the other alleged grounds of unpatentability proffered in the Petition.

It is FURTHER ORDERED that pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is hereby given of the institution of a trial. The trial will commence on the entry date of this decision.

It is FURTHER ORDERED that an initial conference call with the Board is scheduled for 3 PM Eastern Time on February 13, 2014. The parties are directed to the Office Trial Practice Guide, 77 Fed. Reg. 48,756, 48,765-66 (Aug. 14, 2012) for guidance in preparing for the initial conference call, and should be prepared to discuss any proposed changes to

the Scheduling Order entered herewith and any motions the parties anticipate filing during the trial.

Case IPR2013-00473
Patent 8,176,267

For PETITIONER:

Melvin D. Chan
AKA CHAN LLP
mel@akachanlaw.com

For PATENT OWNER:

Joshua A. Griswold
David M. Hoffman
FISH & RICHARDSON P.C.
griswold@fr.com
hoffman@fr.com

US008176267B2

(12) **United States Patent**
Chu et al.

(10) **Patent No.: US 8,176,267 B2**
(45) **Date of Patent: May 8, 2012**

(54) **DATA ACCESSING METHOD FOR FLASH MEMORY STORAGE DEVICE HAVING DATA PERTURBATION MODULE, AND STORAGE SYSTEM AND CONTROLLER USING THE SAME**

(75) Inventors: **Chien-Hua Chu**, Hsinchu County (TW); **Chih-Kang Yeh**, Kinmen County (TW)

(73) Assignee: **Phison Electronics Corp.**, Miaoli (TW)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 874 days.

(21) Appl. No.: **12/210,406**

(22) Filed: **Sep. 15, 2008**

(65) **Prior Publication Data**

US 2010/0011151 A1 Jan. 14, 2010

(30) **Foreign Application Priority Data**

Jul. 9, 2008 (TW) ..... 97125974 A

(51) **Int. Cl.**
G06F 13/00 (2006.01)

(52) **U.S. Cl.** **711/154**; 711/203

(58) **Field of Classification Search** 711/103, 711/154, 203

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2007/0223286	A1*	9/2007	Suda et al.	365/185.29
2007/0233941	A1*	10/2007	Lee et al.	711/103
2009/0055620	A1*	2/2009	Feldman et al.	711/202
2009/0259916	A1*	10/2009	Hsu et al.	714/758
2009/0292862	A1*	11/2009	Kitahara	711/103

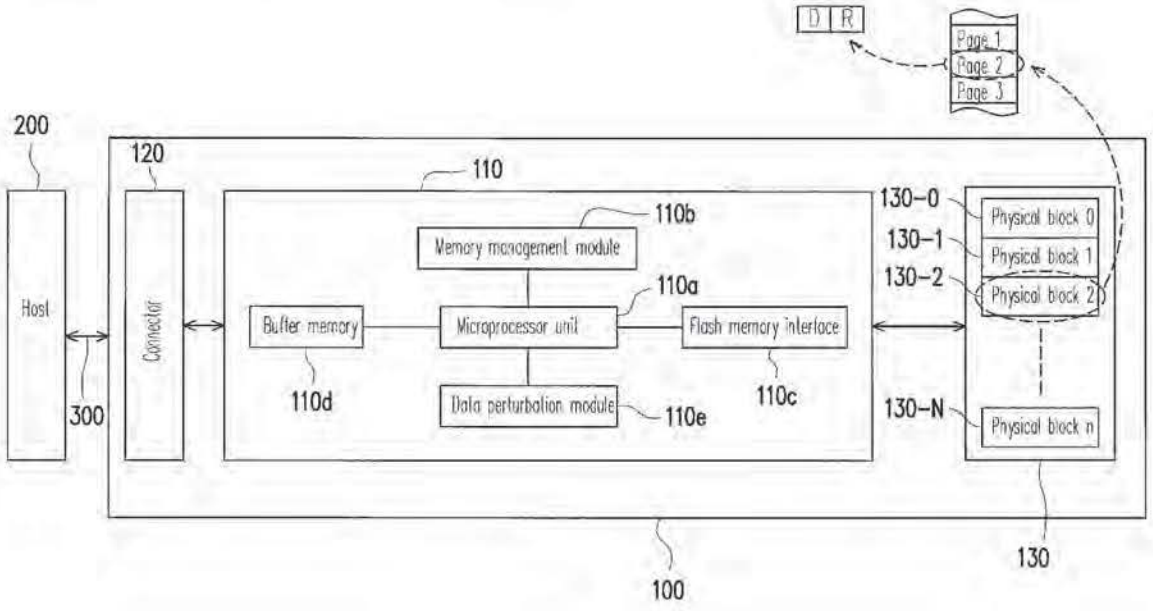
* cited by examiner

Primary Examiner — Reba I Elmore
(74) *Attorney, Agent, or Firm* — J.C. Patents

(57) **ABSTRACT**

A data accessing method, and a storage system and a controller using the same are provided. The data accessing method is suitable for a flash memory storage system having a data perturbation module. The data accessing method includes receiving a read command from a host and obtaining a logical block to be read and a page to be read from the read command. The data accessing method also includes determining whether a physical block in a data area corresponding to the logical block to be read is a new block and transmitting a predetermined data to the host when the physical block corresponding to the logical block to be read is a new block. Thereby, the host is prevented from reading garbled code from the flash memory storage system having the data perturbation module.

25 Claims, 8 Drawing Sheets



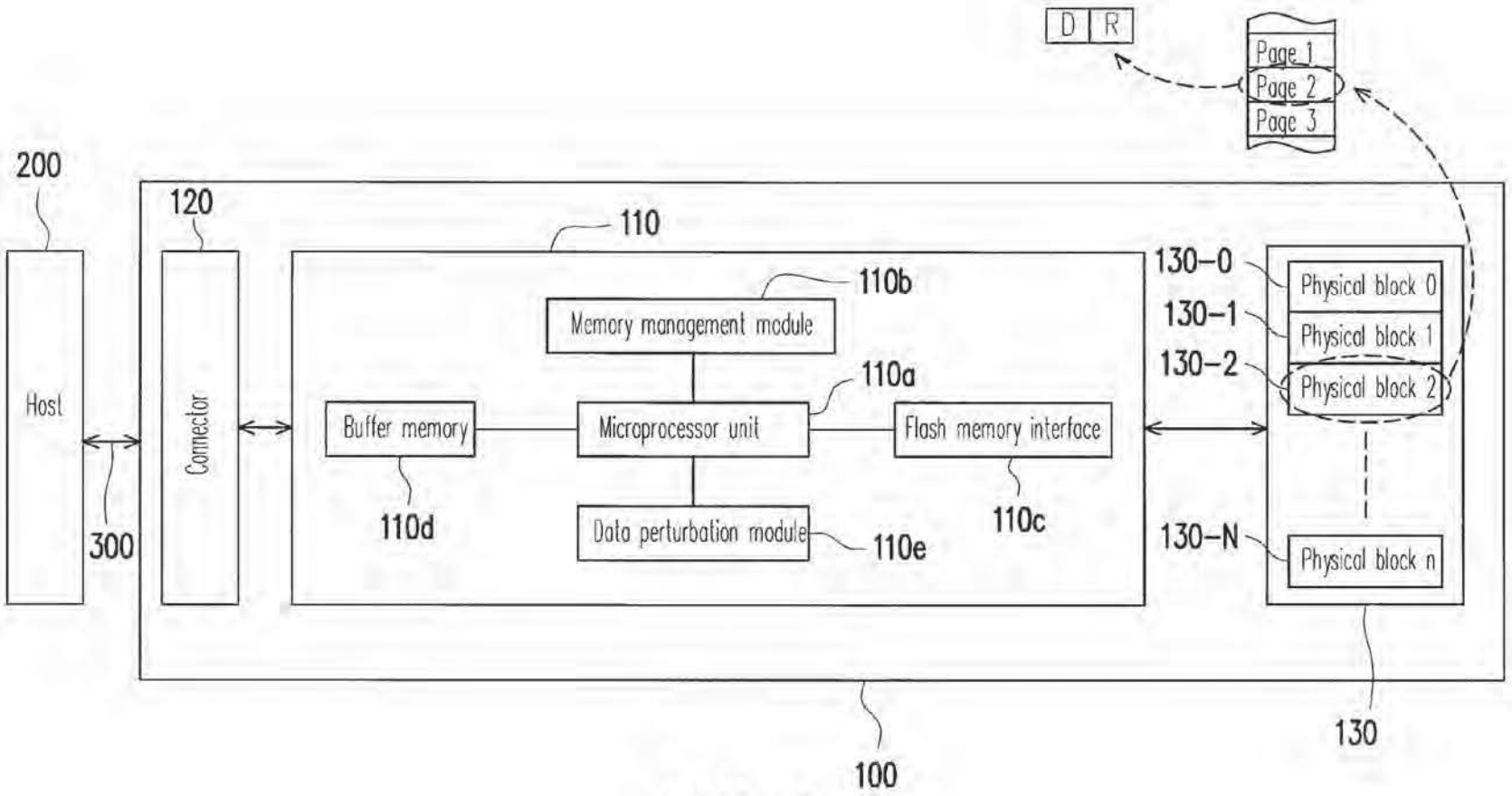

D
R
Page 1
Page 2
Page 3
200
120
110
110b
Memory management module
110a
Host
Connector
Buffer memory
Microprocessor unit
Flash memory interface
110d
Data perturbation module
110e
110c
300
130-0
130-1
130-2
130-N
Physical block 0
Physical block 1
Physical block 2
Physical block n
130
100

FIG. 1

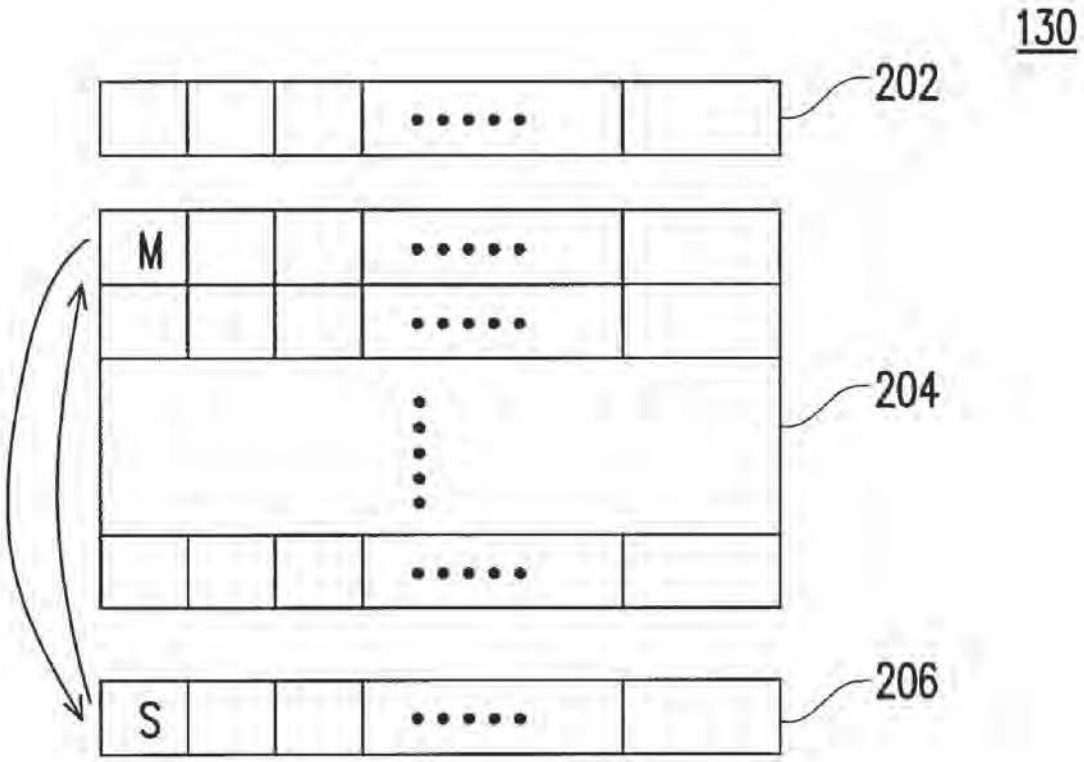


FIG. 2

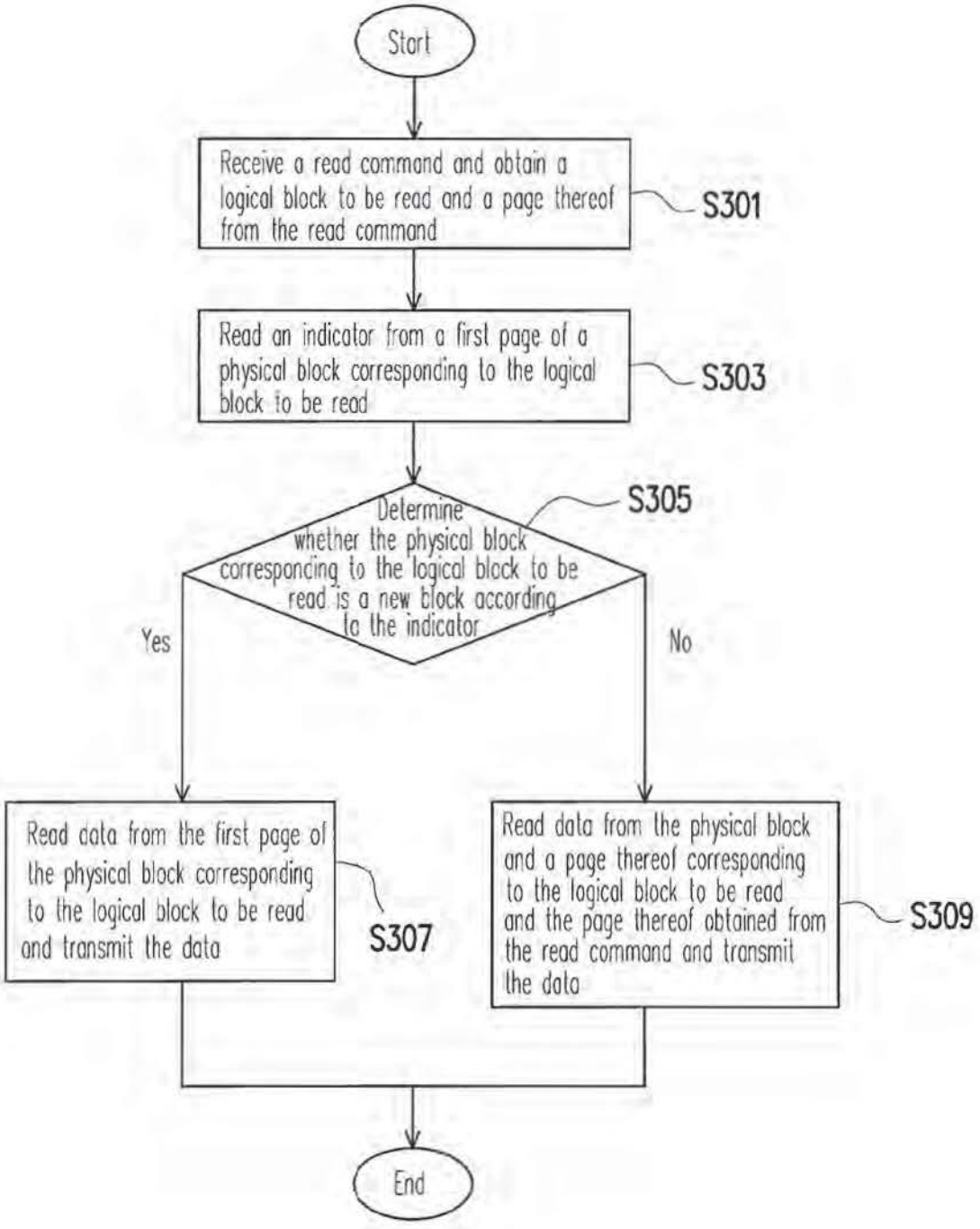

Start
Receive a read command and obtain a logical block to be read and a page thereof from the read command
S301
Read an indicator from a first page of a physical block corresponding to the logical block to be read
S303
Determine whether the physical block corresponding to the logical block to be read is a new block according to the indicator
S305
Yes
No
Read data from the first page of the physical block corresponding to the logical block to be read and transmit the data
S307
Read data from the physical block and a page thereof corresponding to the logical block to be read and the page thereof obtained from the read command and transmit the data
S309
End

FIG. 3A

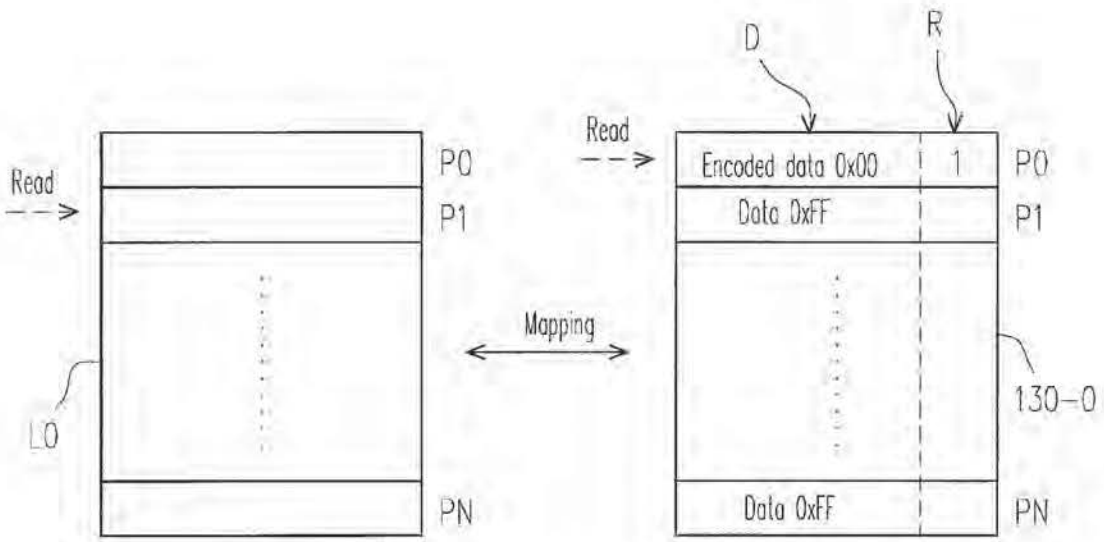

D
R
Read
Read
P0
P1
PN
Encoded data 0x00
1
Data 0xFF
Data 0xFF
Mapping
L0
130-0

FIG. 3B

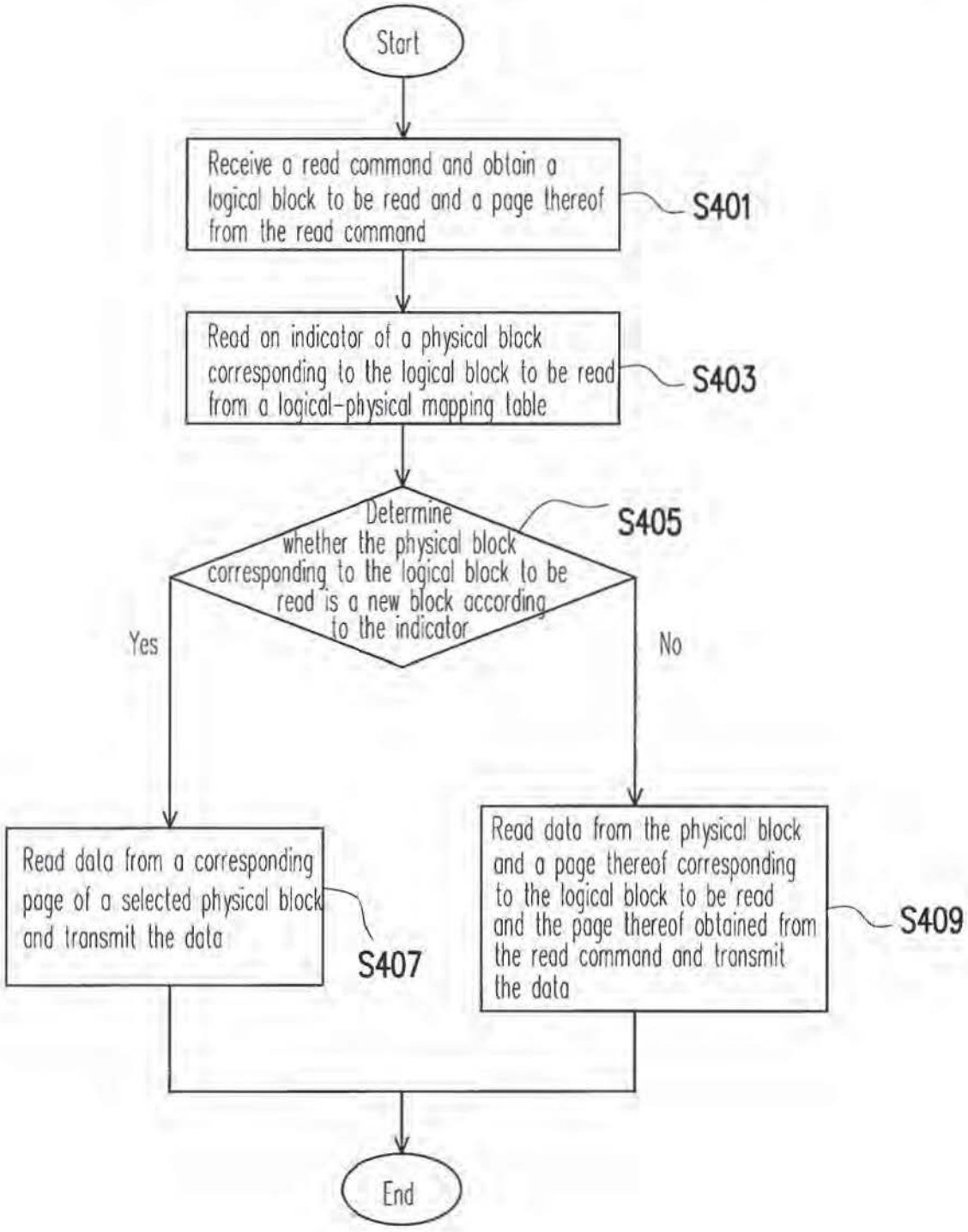

Start
Receive a read command and obtain a logical block to be read and a page thereof from the read command
S401
Read an indicator of a physical block corresponding to the logical block to be read from a logical-physical mapping table
S403
Determine whether the physical block corresponding to the logical block to be read is a new block according to the indicator
S405
Yes
No
Read data from a corresponding page of a selected physical block and transmit the data
S407
Read data from the physical block and a page thereof corresponding to the logical block to be read and the page thereof obtained from the read command and transmit the data
S409
End

FIG. 4A

Logical block	Physical block	Indicator
L0	130-0	1
L1	130-1	1
⋮	⋮	⋮
LK	130-K	1

FIG. 4B

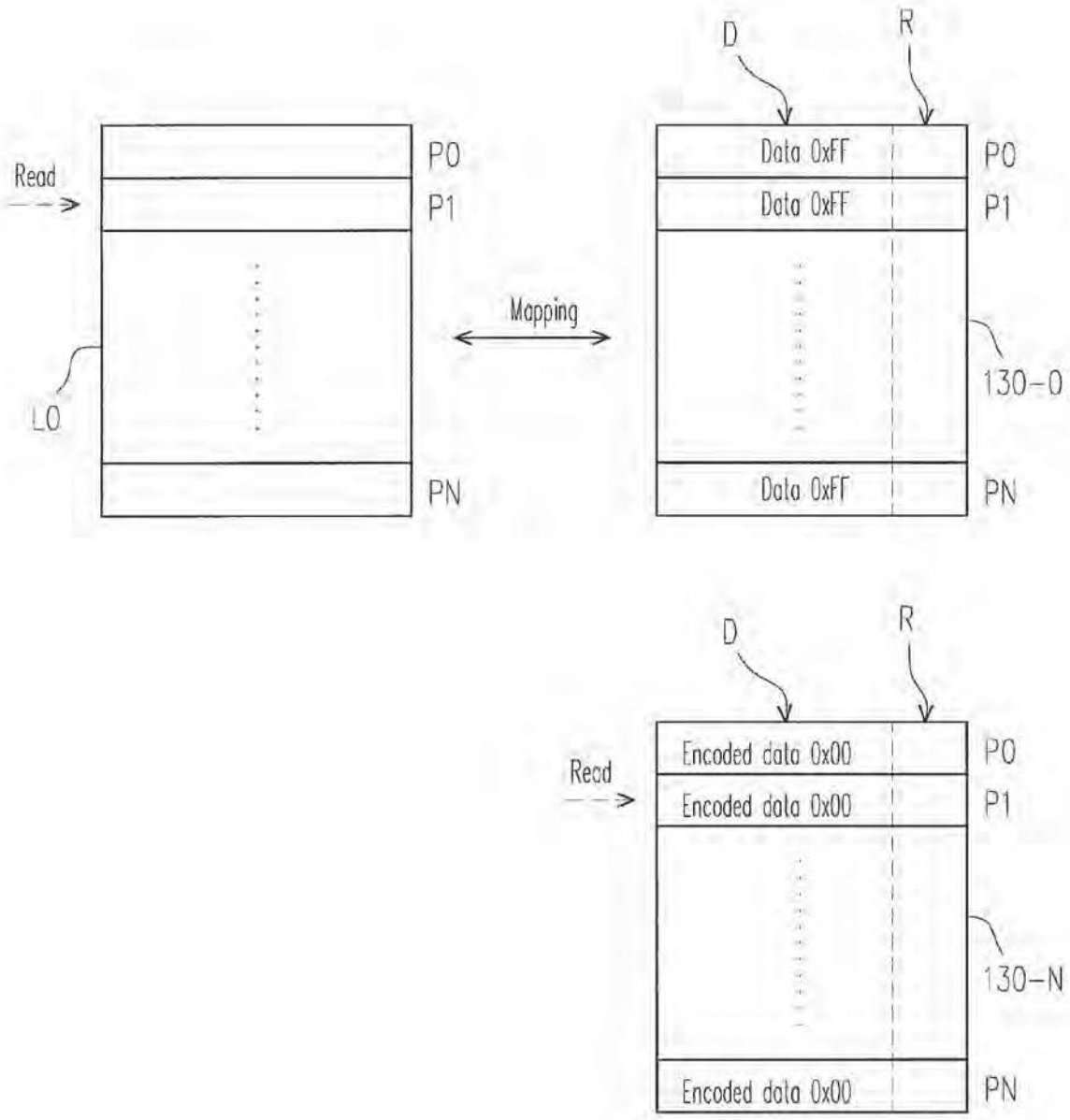

D
R
Read
P0
P1
PN
L0
Mapping
Data 0xFF
Data 0xFF
Data 0xFF
130-0
D
R
Read
Encoded data 0x00
Encoded data 0x00
Encoded data 0x00
130-N

FIG. 4C

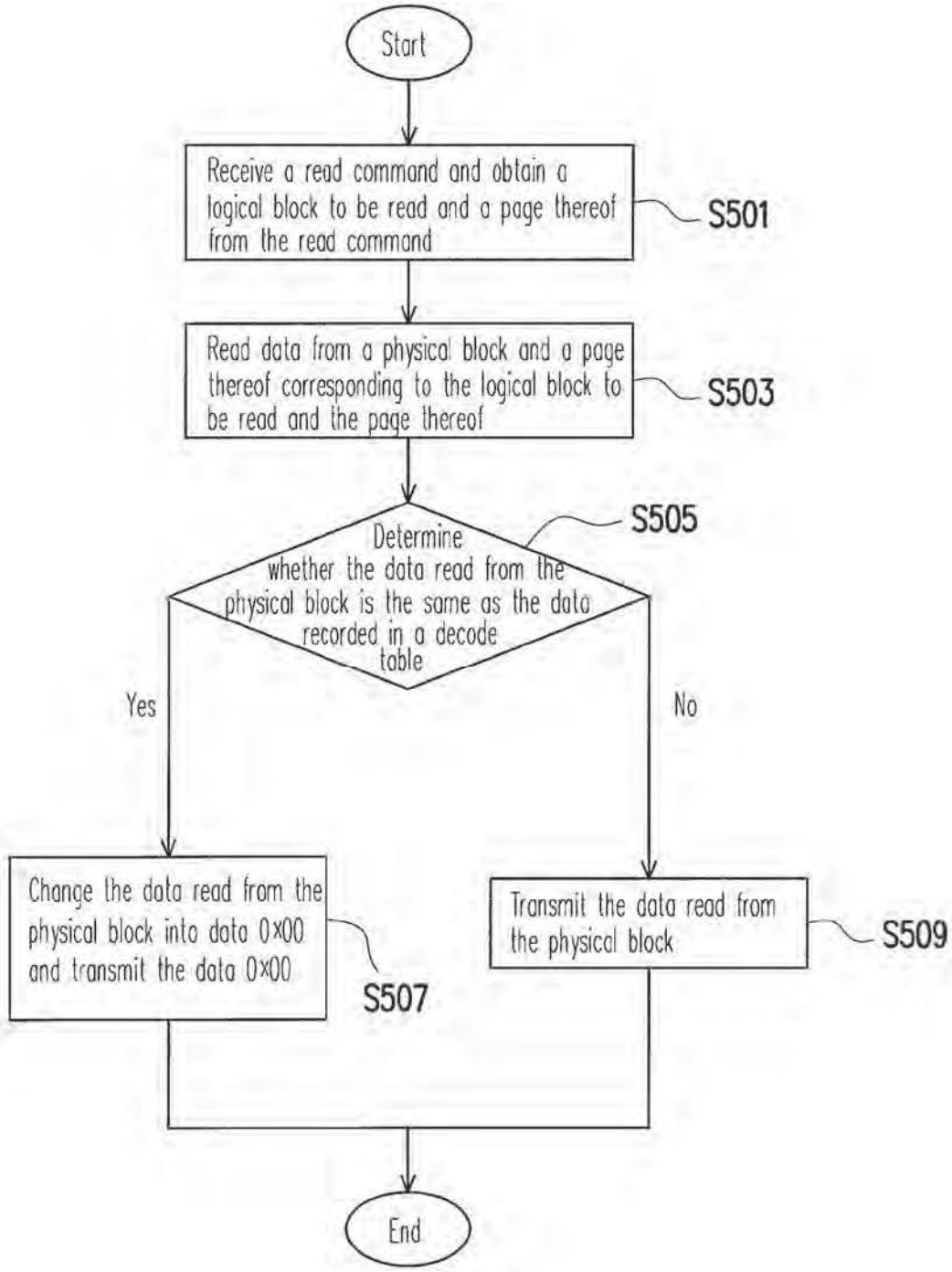

Start
Receive a read command and obtain a logical block to be read and a page thereof from the read command
S501
Read data from a physical block and a page thereof corresponding to the logical block to be read and the page thereof
S503
Determine whether the data read from the physical block is the same as the data recorded in a decode table
S505
Yes
No
Change the data read from the physical block into data 0x00 and transmit the data 0x00
S507
Transmit the data read from the physical block
S509
End

FIG. 5

DATA ACCESSING METHOD FOR FLASH MEMORY STORAGE DEVICE HAVING DATA PERTURBATION MODULE, AND STORAGE SYSTEM AND CONTROLLER USING THE SAME

CROSS-REFERENCE TO RELATED APPLICATION

This application claims the priority benefit of Taiwan application serial no. 97125974, filed on Jul. 9, 2008. The entirety of the above-mentioned patent application is hereby incorporated by reference herein and made a part of specification.

BACKGROUND

1. Technology Field

The present invention generally relates to a data accessing method, and more particularly, to a data accessing method which can prevent a host from reading garbled codes, and a flash memory storage system and a controller using the same.

2. Description of Related Art

Along with the widespread of digital cameras, camera phones, and MP3 in recently years, the consumers' demand to storage media has increased drastically too. Flash memory is one of the most adaptable memories for such battery-powered portable products due to its characteristics such as data non-volatility, low power consumption, small volume, and non-mechanical structure. A memory card is a storage device which uses a NAND flash memory as its storage medium. Memory card has been broadly adopted for storing personal data due to its small volume, large storage capacity, and high portability. Thereby, flash memory has become one of the most focused electronic products in recent years.

Conventionally, the firmware executed by a controller of a flash memory storage system is stored in a programmable read-only memory (PROM) of the controller, and the firmware is loaded into a static random access memory (SRAM) of the controller to be executed during the operation of the flash memory storage system. In order to reduce the size of the storage device and update/modify the firmware conveniently, a technique for directly storing the firmware in a flash memory of a flash memory storage system and loading the firmware into the controller only when the controller is about to work has been developed. According to this technique, a data perturbation module (or an encoding module) is usually implemented and a data is encoded into a garbled code before it is transmitted to the flash memory. Besides, after the data is read from the flash memory, the data is decoded to restore the original data. As described above, the security of the firmware or other important data transmitted between the controller and the flash memory is protected.

However, a card activation process has to be performed to a new flash memory in a flash memory storage device when the flash memory storage device is just manufactured, wherein the card activation process is to initialize each new block in the flash memory (i.e., setting the data at each page address in each block to be 0xFF). In particular, this initialization action is not a write command and accordingly the data is not encoded by the data perturbation module. However, when an end-user system issues a read command to a new block in the new flash memory storage device, the data 0xFF stored in the new block is decoded by the data perturbation module and accordingly the end-user system receives an unrecognizable garbled code. Accordingly, a data accessing method which can prevent a host from reading garbled code from a flash memory storage device having a data perturbation module is required.

SUMMARY

Accordingly, the present invention is directed to a data accessing method which can prevent a host from reading garbled codes from a flash memory storage system having a data perturbation module.

The present invention is directed to a controller, wherein a memory management module manages and reads new blocks of a flash memory through foregoing data accessing method such that a host is prevented from reading garbled codes from a flash memory storage system having a data perturbation module.

The present invention is further directed to a storage system, wherein a controller manages and reads new blocks of a flash memory through foregoing data accessing method such that a host is prevented from reading garbled codes from a flash memory storage system having a data perturbation module.

The present invention provides a data accessing method suitable for accessing a flash memory storage device having a data perturbation module, wherein a flash memory of the flash memory storage device has a plurality of physical blocks, and these physical blocks are grouped into at least a data area and a spare area. The data accessing method includes receiving a read command from a host and obtaining a logical block to be read and a page to be read from the read command. The data accessing method also includes determining whether a physical block in the data area corresponding to the logical block to be read is a new block and transmitting a predetermined data to the host when the physical block corresponding to the logical block to be read is a new block.

According to an embodiment of the present invention, the predetermined data is 0x00 or 0xFF.

According to an embodiment of the present invention, the data accessing method further includes recording an indicator for each physical block to indicate that the physical block is a new block during a card activation process performed to the flash memory storage device. The step for determining whether a physical block in the data area corresponding to the logical block to be read is a new block includes determining whether the physical block in the data area corresponding to the logical block to be read is a new block according to the indicator. It is determined that the physical block corresponding to the logical block to be read is a new block if the physical block corresponding to the logical block to be read has the indicator.

According to an embodiment of the present invention, the data accessing method further includes writing the predetermined data into a predetermined page of each physical block during the card activation process performed to the flash memory storage device, and the step for transmitting the predetermined data to the host when the physical block corresponding to the logical block to be read is a new block includes reading data from the predetermined page of the physical block corresponding to the logical block to be read.

According to an embodiment of the present invention, the predetermined page is a first page of the physical block.

According to an embodiment of the present invention, the data accessing method further includes writing the predetermined data into all the pages of one of the physical blocks in the spare area during the card activation process performed to the flash memory storage device, and the step for transmitting the predetermined data to the host when the physical block

corresponding to the logical block to be read is a new block includes reading data from a page corresponding to the page to be read in the one of the physical blocks.

According to an embodiment of the present invention, the step for transmitting the predetermined data to the host when the physical block corresponding to the logical block to be read is a new block includes changing the data read from the physical block corresponding to the logical block to be read into the predetermined data through hardware or firmware.

According to an embodiment of the present invention, the data accessing method further includes decoding the data 0xFF in each physical block by using the data perturbation module and recording the decoded data 0xFF into a decode table during the card activation process performed to the flash memory storage device. The step for determining whether the physical block in the data area corresponding to the logical block to be read is a new block includes comparing the data read from the physical block corresponding to the logical block to be read with the data recorded in the decode table. It is determined that the physical block corresponding to the logical block to be read is a new block if the data read from the physical block corresponding to the logical block to be read is the same as the data recorded in the decode table.

According to an embodiment of the present invention, the data accessing method further includes recording the indicator in a logical-physical mapping table.

According to an embodiment of the present invention, the data accessing method further includes recording the indicator in a redundant area of each physical block.

The present invention provides a flash memory storage system including a flash memory, a connector, and a controller. The flash memory is used for storing data, wherein the flash memory includes a plurality of physical blocks, and these physical blocks are grouped into at least a data area and a spare area. The connector is used for connecting to a host. The controller is electrically connected to the flash memory and the connector, and the controller includes a microprocessor unit, a data perturbation module, a flash memory interface, a buffer memory, and a memory management module. The microprocessor unit obtains a logical block to be read and a page to be read from a read command received from a host. The data perturbation module is electrically connected to the microprocessor unit for decoding a data read from the flash memory. The flash memory interface is electrically connected to the microprocessor unit for accessing the flash memory. The buffer memory is electrically connected to the microprocessor unit for temporarily storing data. The memory management module is electrically connected to the microprocessor unit, and the memory management module determines whether a physical block in the data area corresponding to the logical block to be read is a new block and transmits a predetermined data to the host when the physical block corresponding to the logical block to be read is a new block.

According to an embodiment of the present invention, the predetermined data is 0x00 or 0xFF.

According to an embodiment of the present invention, the memory management module records an indicator for each physical block to indicate that the physical block is a new block during a card activation process performed to the flash memory storage device, and the memory management module determines whether the physical block in the data area corresponding to the logical block to be read is a new block according to the indicator.

According to an embodiment of the present invention, the memory management module writes the predetermined data into a predetermined page of each physical block during the card activation process, and the memory management module reads data from the predetermined page of the physical block corresponding to the logical block to be read when the physical block corresponding to the logical block to be read is a new block.

According to an embodiment of the present invention, the predetermined page is a first page of the physical block.

According to an embodiment of the present invention, the memory management module writes the predetermined data into all the pages of one of the physical blocks in the spare area during the card activation process, and the memory management module reads data from a page corresponding to the page to be read in the one of the physical blocks when the physical block corresponding to the logical block to be read is a new block.

According to an embodiment of the present invention, the memory management module changes the data read from the physical block corresponding to the logical block to be read into the predetermined data when the physical block corresponding to the logical block to be read is a new block.

According to an embodiment of the present invention, the memory management module decodes the data 0xFF in each physical block by using the data perturbation module and records the decoded data 0xFF into a decode table during the card activation process, and the memory management module determines that the physical block corresponding to the logical block to be read is a new block when the data read from the physical block corresponding to the logical block to be read is the same as the data recorded in the decode table.

According to an embodiment of the present invention, the memory management module records the indicator in a logical-physical mapping table.

According to an embodiment of the present invention, the memory management module records the indicator in a redundant area of each physical block.

According to an embodiment of the present invention, the flash memory storage device is a flash drive, a memory card, or a solid state drive (SSD).

In the present invention, whether a physical block is a new block is determined before a data is transmitted to a host, and a predetermined data is transmitted to the host when the physical block is a new block. Thereby, the host is prevented from reading garbled code.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings are included to provide a further understanding of the invention, and are incorporated in and constitute a part of this specification. The drawings illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention.

FIG. **1** is a schematic block diagram of a flash memory storage system according to an embodiment of the present invention.

FIG. **2** is a detailed block diagram of a flash memory and operations thereof according to an embodiment of the present invention.

FIG. **3**A is a flowchart of a data accessing method according to a first embodiment of the present invention.

FIG. **3**B is a diagram illustrating how a host changes to read another page according to the first embodiment of the present invention.

FIG. **4**A is a flowchart of a data accessing method according to a second embodiment of the present invention.

FIG. **4**B illustrates an example of a logical-physical mapping table according to the second embodiment of the present invention.

FIG. **4**C is a diagram illustrating how a host changes to read another physical block according to the second embodiment of the present invention.

FIG. **5** is a flowchart of a data accessing method according to a third embodiment of the present invention.

DESCRIPTION OF THE EMBODIMENTS

Reference will now be made in detail to the present preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers are used in the drawings and the description to refer to the same or like parts.

FIG. **1** is a schematic block diagram of a flash memory storage system according to an embodiment of the present invention. Referring to FIG. **1**, the flash memory storage system **100** includes a controller **110**, a connector **120**, and a flash memory **130**.

The flash memory storage system **100** usually works together with a host **200** to allow the host **200** to write data into or read data from the flash memory storage system **100**. In the present embodiment, the flash memory storage system **100** is a solid state drive (SSD). However, in another embodiment of the present invention, the flash memory storage system **100** may also be a memory card or a flash drive.

The controller **110** executes a plurality of commands implemented as hardware or firmware and coordinates with the connector **120** and the flash memory **130** to perform various data operations, such as data storing, reading, and erasing, etc. The controller **110** includes a microprocessor unit **110***a*, a memory management module **110***b*, a flash memory interface **110***c*, a buffer memory **110***d*, and a data perturbation module **110***e*.

The microprocessor unit **110***a* coordinates with the memory management module **110***b*, the flash memory interface **110***c*, the buffer memory **110***d* and a data perturbation module **110***e* to perform various operations of the flash memory storage system **100**. For example, the microprocessor unit **110***a* works together with the memory management module **110***b*, the flash memory interface **110***c* the buffer memory **110***d* and a data perturbation module **110***e* to read and write data according to a read command or a write command it receives.

The memory management module **110***b* is electrically connected to the microprocessor unit **110***a*. The memory management module **110***b* manages the flash memory **130**. For example, the memory management module **110***b* has machine instructions for executing wear levelling, managing bad blocks, and maintaining a mapping table, etc. In particular, the memory management module **110***b* executes a data accessing method as described below with reference to several embodiments of the present invention.

It should be mentioned that in the present embodiment, the memory management module **110***b* is implemented as a hardware. However, in another embodiment of the present invention, the memory management module **110***b* may also be implemented as a firmware. When the memory management module **110***b* is implemented as a firmware, the memory management module **110***b* has a plurality of machine instructions which can be executed by the microprocessor unit **110***a*, and when the flash memory storage system **100** is in operation, the memory management module **110***b* is loaded into the buffer memory **110***d* to be executed by the microprocessor unit **110***a*.

The flash memory interface **110***c* is electrically connected to the microprocessor unit **110***a* for accessing the flash memory **130**. Namely, data to be written into the flash memory **130** by the host **200** is converted by the flash memory interface **110***c* into a format acceptable to the flash memory **130**.

The buffer memory **110***d* is electrically connected to the microprocessor unit **110***a* for temporarily storing system data (such as a logical-physical mapping table) or data to be read or written by the host **200**. In the present embodiment, the buffer memory **110***d* is a static random access memory (SRAM). However, the present invention is not limited thereto, and a dynamic random access memory (DRAM), a magnetoresistive random access memory (MRAM), a phase change random access memory (PRAM), or other suitable memories may also be applied in the present invention.

The data perturbation module **110***e* is electrically connected to the microprocessor unit **110***a* for converting a sequence data into a non-sequence data or a non-sequence data into a sequence data. To be specific, to ensure data security, in the present embodiment, all the data transmitted when the controller **110** writes data into or read data from the flash memory **130** is encoded or decoded by the data perturbation module **110***e*.

In an embodiment of the present invention, the data perturbation module **110***e* switches data of at least two bytes in a received data. For example, when the sequence data it receives is "01 02 03 04 05 06 07 08 11 12 13 14 15 16 17 18", the data perturbation module **110***e* encodes the sequence data into "05 06 07 08 01 02 03 04 15 16 17 18 11 12 13 14". Contrarily, when the non-sequence data it receives is "05 06 07 08 01 02 03 04 15 16 17 18 11 12 13 14", the data perturbation module **110***e* decodes the non-sequence data into "01 02 03 04 05 06 07 08 11 12 13 14 15 16 17 18". However, foregoing example of data switching is not intended to restricting the scope of the present invention, and other data switching or re-arranging pattern can be easily implemented by those skilled in the art according to the present disclosure.

In the present embodiment, the data perturbation module **110***e* scrambles a data by changing the sequence of the data; however, in another embodiment of the present invention, the data perturbation module **110***e* may also scramble a data by inverting each bit of the data (for example, change "0" into "1" or "1" into "0") or by using an algorithm.

In addition, even though not shown in the present embodiment, the controller **110** may further include general function modules for controlling the flash memory, such as an error correction module and a power management module.

The connector **120** connects to the host **200** through a bus **300**. In the present embodiment, the connector **120** is a PCI Express connector. However, the present invention is not limited thereto, and the connector **120** may also be a USB connector, an IEEE 1394 connector, a SATA connector, a MS connector, a MMC connector, a SD connector, a CF connector, an IDE connector, or other suitable data transmission connectors.

The flash memory **130** is electrically connected to the controller **110** for storing data. The flash memory **130** is substantially divided into a plurality of physical blocks **130-0**~**130-N**. Generally speaking, data in a flash memory is erased in unit of physical blocks. Namely, each physical block contains the smallest number of memory cells which are erased together. Each physical block is usually divided into a plurality of pages. Page is usually the smallest programming unit. However, it should be noted that in some different flash memory designs, the smallest programming unit may also be sector, namely, a page has a plurality of sectors and each sector is served as the smallest programming unit. In other words, page is served as the smallest unit for reading and

writing data. A page is usually divided into a user data area D and a redundant area R, wherein the user data area D is used for storing user data, and the redundant area R is used for storing system data (for example, an error correcting code (ECC)).

Generally speaking, the user data area D has 512 bytes and the redundant area R has 16 bytes in order to correspond to the size of sectors in a disk drive. In other words, a page is a sector. However, a page may also be composed of a plurality of sectors. For example, a page may include four sectors.

Generally speaking, a physical block can be composed of any number of pages, such as 64 pages, 128 pages, and 256 pages, etc. The physical blocks **130-0**~**130-N** are usually grouped into several zones. By managing the operations of a memory based on zones, parallelism of the operations can be increased and the management can be simplified.

The operation of the flash memory **130** will be described according to the present invention with reference to accompanying drawings. It should be understood that the terms "select", "move", and "substitute" used in following description only refer to logical operations performed to the flash memory **130**. In other words, the physical positions of the blocks in the flash memory are not changed; instead, these blocks in the flash memory **130** are only operated logically.

FIG. **2** is a detailed block diagram of the flash memory **130** and the operations thereof according to an embodiment of the present invention.

Referring to FIG. **2**, in the present embodiment, in order to program (i.e., write and erase) the flash memory **130** more efficiently, the physical blocks **130-1**~**130-N** in the flash memory **130** are logically grouped into a system area **202**, a data area **204**, and a spare area **206**. Generally speaking, more than 90% of the physical blocks in the flash memory **130** belong to the data area **204**.

Physical blocks in the system area **202** are used for recording system data, such as number of zones in the flash memory **130**, number of physical blocks in each zone, number of pages in each physical block, and the logical-physical mapping table, etc.

Physical blocks in the data area **204** are used for storing user data, and these physical blocks are actually the blocks corresponding to the logical block addresses (LBA) operated by the host **200**.

Physical blocks in the spare area **206** are used for substituting the physical blocks in the data area **204**. Thus, the physical blocks in the spare area **206** are blank or available blocks, namely, no data is recorded in these blocks or data recorded in these blocks has been marked as invalid data. To be specific, an erasing operation has to be performed before writing data to an address in which data has been recorded. However, as described above, data is written into a flash memory in unit of pages while erased from the same in unit of blocks. Since the erase unit is larger than the write unit, valid pages in a physical block have to be copied to another physical block before data in the physical block is erased. Accordingly, to write a new data into a physical block M in the data area **204** which already contains a data, a physical block S is first selected from the spare area **206**, and the valid data in the physical block M is copied to the physical block S and the new data is also written into the physical block S. After that, the physical block M is erased and moved to the spare area **206**, and at the same time, the physical block S is moved to the data area **204**. It should be understood that moving the physical block M to the spare area **206** and the physical block S to the data area **204** is to link the physical block M to the spare area **206** and the physical block S to the data area **204** logically. It should be understood by those having ordinary knowledge in the art that the logical relationship between blocks in the data area **204** can be maintained through a logical-physical mapping table.

A card activation process is usually performed to the flash memory storage system **100** right after the flash memory storage system **100** is manufactured. During the card activation process, all the physical blocks **130-0**~**130-N** of the flash memory **130** are initialized to set the data therein to be 0xFF. As described above, the physical blocks **130-0**~**130-N** of the flash memory **130** are grouped into the system area, the data area, and the spare area, wherein the physical blocks in the data area are blocks containing valid data and to be accessed by the host **200** (i.e., blocks corresponding to the logical blocks), and the physical blocks in the spare area are used for substituting the blocks in the data area while executing write commands. Thus, the problem of getting garbled code just is occurred in the case of reading physical blocks (i.e., new blocks) which are just initialized (i.e., data stored therein is 0xFF). That is, when the physical blocks in the data area and the physical blocks in the spare area once are written data with the above-mentioned substituting operation, the problem of getting garbled code is not occurred. According to the data accessing method provided by the present invention, whether a physical block is a new block is determined before a data is transmitted to the host **200**, and if the physical block is a new block, a predetermined data (for example, 0x00 or 0xFF) is transmitted to the host **200** so as to prevent the host **200** from receiving garbled code. Embodiments of the present invention will be described below with reference to accompanying drawings.

First Embodiment

As described above, a card activation process is performed to the flash memory storage system **100** after the flash memory storage system **100** is manufactured. Herein, the memory management module **110***b* uses one bit in a redundant area R of a first page in each of the physical blocks for indicating that the physical block is a new block (i.e., no data has been written therein). For example, the memory management module **110***b* records "1" in this bit for indicating that the physical block is a new block and records "0" for indicating that the physical block is not a new block. Meanwhile, during the card activation process, the memory management module **110***b* writes data 0x00 into a data area D of the first page in each of the physical blocks. In particular, the data 0x00 is encoded by the data perturbation module **110***e* and written as a garbled code into the data area D of the first page in each physical block. It should be understood that herein the data 0x00 (i.e., the predetermined data) is written into the first page of the physical block; however, the present invention is not limited thereto, and in another embodiment of the present invention, the predetermined data may also be written into another page of the physical block.

It should be mentioned that subsequently, when a data is written into a physical block which is indicated as a new block during the operation of the flash memory storage system **100**, the memory management module **110***b* changes the indicator of the physical block to indicate that the physical block is not a new block. In particular, if the data is written into only some pages of the physical block which is indicated as a new block, data 0x00 is written into the other pages to make sure that all the pages in the physical block have been written.

In an embodiment of the present invention, when the host **200** desires to read data from the flash memory storage system **100**, the memory management module **110***b* determines whether the physical block corresponding to the page the host

200 desires to read is a new block. If the page to be read is in a new block, the memory management module **110***b* reads the first page of the physical block and transmits the data therein to the host **200**. To be more specific, if the page to be read is in a new block, then the data stored in this page is 0xFF, and a garbled code is produced when the data 0xFF is decoded by the data perturbation module **110***e*, and accordingly the host **200** receives the unrecognizable garbled code from this page. Thus, in the present embodiment, if the physical block corresponding to the page to be read is a new block, the memory management module **110***b* reads data from the first page of the physical block and transmits the data to the host, wherein because the data in the first page of the physical block is stored as the data 0x00 encoded by the data perturbation module **110***e*, data 0x00 is obtained when the data read from the first page of the physical block is decoded by the data perturbation module **110***e*. Accordingly, the host **200** is prevented from reading garbled code.

In the present embodiment, the indicator indicating whether a physical block is a new block is recorded in the redundant area R of the first page in the physical block. However, the present invention is not limited thereto, and in another embodiment of the present invention, an indicator indicating whether a physical block corresponding to a logical block is a new block may also be recorded in a logical-physical mapping table.

FIG. **3**A is a flowchart of a data accessing method according to the first embodiment of the present invention.

Referring to FIG. **3**A, in step S**301**, the flash memory storage system **100** receives a read command from the host **200**, and the microprocessor unit **110***a* obtains a logical block to be read and a page thereof from the read command. Then, in step S**303**, the memory management module **110***b* reads the indicator from the first page of a physical block corresponding to the logical block to be read, and in step S**305**, the memory management module **110***b* determines whether the physical block corresponding to the logical block to be read is a new block according to the indicator.

If it is determined in step S**305** that the physical block corresponding to the logical block to be read is a new block, then in step S**307**, the memory management module **110***b* reads data from the first page of the physical block corresponding to the logical block to be read and transmits the data to the host **200**. For example, as shown in FIG. **3**B, when the host **200** desires to read a page P**1** of a logical block L**0**, the memory management module **110***b* determines that the physical block **130-0** corresponding to the logical block L**0** is a new block according to the indicator "1" in the page P**0** of the physical block **130-0**. Thus, the memory management module **110***b* reads data from the page P**0** of the physical block **130-0** instead of the page P**1** of the physical block **130-0**.

If it is determined in step S**305** that the physical block corresponding to the logical block to be read is not a new block, then in step S**309**, the memory management module **110***b* reads data from the physical block and the page thereof corresponding to the logical block and the page thereof obtained from the read command and transmits the data to the host **200**.

Second Embodiment

As described above, in the first embodiment, when it is determined that the physical block to be read is a new block, data 0x00 previously written into the physical block is read by changing the page to be read in the physical block so as to prevent the host from reading garbled code. However, this purpose may also be achieved by changing the physical block to be read.

As described above, during the card activation process, all the physical blocks of the flash memory **130** are initialized and data stored therein are set to be 0xFF. Besides, according to the present embodiment, the memory management module **110***b* indicates that the physical block corresponding to each logical block is a new block in the logical-physical mapping table. For example, "1" is recorded in one bit of each record in the logical-physical mapping table for indicating that the physical block is a new block (as shown in FIG. **4**B), and when subsequently a data is written into the physical block which is indicated as a new block during the operation of the flash memory storage system **100**, the indicator is changed to "0" to indicate that the physical block is not a new block. Meanwhile, in the present embodiment, the memory management module **110***b* selects a physical block (for example, the physical block **130**-N in FIG. **4**C) from the spare area **206** of the flash memory and writes data 0x00 into all the pages of the selected physical block. In particular, the data 0x00 is encoded by the data perturbation module **110***e* before it is written into all the pages of the selected physical block.

In the present embodiment, when the host **200** desires to read data from the flash memory storage system **100**, the memory management module **110***b* determines whether a physical block corresponding to the page to be read by the host **200** is a new block, and if the physical block corresponding to the page to be read is a new block, the memory management module **110***b* reads data from the corresponding page in the selected physical block and transmits the data to the host **200**. In the present embodiment, if the physical block corresponding to a page to be read is a new block, the memory management module **110***b* reads data from a physical block which is previously written with data 0x00 and transmits the data to the host. Thereby, the host is prevented from reading garbled code.

FIG. **4**A is a flowchart of a data accessing method according to the second embodiment of the present invention.

Referring to FIG. **4**A, in step S**401**, the flash memory storage system **100** receives a read command from the host **200**, and the microprocessor unit **110***a* obtains a logical block to be read and a page thereof from the read command. Then, in step S**403**, the memory management module **110***b* reads an indicator of a physical block corresponding to the logical block to be read from the logical-physical mapping table, and in step S**405**, the memory management module **110***b* determines whether the physical block corresponding to the logical block to be read is a new block according to the indicator.

If the memory management module **110***b* determines that the physical block corresponding to the logical block to be read is a new block in step S**405**, then in step S**407**, the memory management module **110***b* reads data from the corresponding page in the selected physical block and transmits the data to the host. For example, as shown in FIG. **4**C, when the host **200** is about to read data from the page P**1** of the logical block L**0**, the memory management module **110***b* determines that the physical block **130-0** is a new block according to the indicator "1" in the logical-physical mapping table. Thus, the memory management module **110***b* reads data from the page P**1** of the physical block **130**-N which is previously written with data 0x00 instead of the page P**1** of the physical block **130-0**.

If the memory management module **110***b* determines that the physical block corresponding to the logical block to be read is not a new block in step S**405**, then in step S**409**, the memory management module **110***b* reads data from the

physical block and the page thereof corresponding to the logical block to be read and the page thereof obtained from the read command and transmits the data to the host **200**.

Third Embodiment

In the first embodiment and the second embodiment described above, whether a physical block to be read is a new block is determined according to an indicator recorded in the redundant area of a first page in the physical block or the logical-physical mapping table. Additionally, whether the physical block is a new block may also be determined according to the data read from the physical block.

In the present embodiment, during the card activation process of the flash memory storage system **100**, the memory management module **110***b* decodes the data 0xFF in each initialized physical block by using the data perturbation module **110***e* and records the decoded data in a decode table. When subsequently the controller **110** reads data from a physical block according to a read command received from the host **200**, the memory management module **110***b* can determine whether the physical block is a new block by comparing the data read from the physical block and the data recorded in the decode table.

In addition, in the first embodiment and the second embodiment described above, the previously written data 0x00 is read by respectively changing the page and the physical block to be read so as to prevent the host from reading the garbled code. However, in the present embodiment, the memory management module **110***b* may respond to the read command by directly transmitting the data 0x00 to the host when it determines that the physical block corresponding to the logical block to be read is a new block, wherein the instruction for changing the data in the memory management module **110***b* may be implemented as a hardware or a firmware.

FIG. **5** is a flowchart of a data accessing method according to the third embodiment of the present invention.

Referring to FIG. **5**, in step **S501**, the flash memory storage system **100** receives a read command from the host **200**, and the microprocessor unit **110***a* obtains a logical block to be read and a page thereof from the read command. Then, in step **S503**, the memory management module **110***b* reads data from a physical block and a page thereof corresponding to the logical block to be read and the page thereof. Next, in step **S505**, the memory management module **110***b* determines whether the data read from the physical block is the same as the data in the decode table, wherein the memory management module **110***b* determines that the physical block corresponding to the logical block to be read is a new block if the data read from the physical block is the same as the data in the decode table.

If the comparison in step **S505** shows that the data read from the physical block is the same as the data in the decode table, then in step **S507**, the data read from the physical block is changed to 0x00, and the data 0x00 is transmitted to the host **200**.

If the comparison in step **S505** shows that the data read from the physical block is different from the data in the decode table, then in step **S509**, the data read from the physical block is transmitted to the host **200**.

In overview, according to the data accessing method provided by the present invention, whether a physical block is a new block is determined before a data is transmitted to an end-user system, and if the physical block is a new block, data 0x00 is transmitted to the end-user system by changing the page or physical block to be read or directly producing the data 0x00. Thereby, the end-user system is prevented from reading garbled codes from a new block in a flash memory storage system having a data perturbation module.

It will be apparent to those skilled in the art that various modifications and variations can be made to the structure of the present invention without departing from the scope or spirit of the invention. In view of the foregoing, it is intended that the present invention cover modifications and variations of this invention provided they fall within the scope of the following claims and their equivalents.

What is claimed is:

1. A data accessing method, suitable for a flash memory storage device having a data perturbation module, wherein a flash memory of the flash memory storage device has a plurality of physical blocks, and the physical blocks are grouped into at least a data area and a spare area, the data accessing method comprising:

receiving a read command from a host, and obtaining a logical block to be read and a page to be read from the read command;

determining whether a physical block in the data area corresponding to the logical block to be read is a new block;

transmitting a predetermined data to the host when the physical block corresponding to the logical block to be read is the new block; and

decoding data read from the physical block corresponding to the logical block to be read by the data perturbation and transmitting the decoded data to the host when the physical block corresponding to the logical block to be read is not the new block.

2. The data accessing method according to claim **1**, wherein the predetermined data is 0x00 or 0xFF.

3. The data accessing method according to claim **1**, further comprising recording an indicator for each of the physical blocks to indicate that the physical block is the new block during a card activation process performed to the flash memory storage device, wherein the step for determining whether the physical block in the data area corresponding to the logical block to be read is the new block comprises:

determining whether the physical block in the data area corresponding to the logical block to be read is the new block according to the indicator, wherein the physical block in the data area corresponding to the logical block to be read is determined to be the new block if the physical block corresponding to the logical block to be read has the indicator.

4. The data accessing method according to claim **3**, further comprising writing the predetermined data into a predetermined page of each of the physical blocks during the card activation process performed to the flash memory storage device, wherein the step for transmitting the predetermined data to the host when the physical block corresponding to the logical block to be read is the new block comprises reading data from the predetermined page of the physical block corresponding to the logical block to be read.

5. The data accessing method according to claim **4**, wherein the predetermined page is a first page of the physical block.

6. The data accessing method according to claim **3**, further comprising writing the predetermined data into all the pages of one of the physical blocks in the spare area during the card activation process performed to the flash memory storage device, wherein the step for transmitting the predetermined data to the host when the physical block corresponding to the logical block to be read is the new block comprises reading data from a page corresponding to the page to be read in the one of the physical blocks.

7. The data accessing method according to claim **3**, wherein the step for transmitting the predetermined data to the host when the physical block corresponding to the logical block to be read is the new block comprises changing the data

read from the physical block corresponding to the logical block to be read into the predetermined data by using a hardware or a firmware.

8. The data accessing method according to claim **1**, further comprising decoding the data 0xFF in each of the physical blocks by using the data perturbation module and recording the decoded data 0xFF into a decode table during the card activation process performed to the flash memory storage device, wherein the step for determining whether the physical block in the data area corresponding to the logical block to be read is the new block comprises:

comparing the data read from the physical block corresponding to the logical block to be read with the data recorded in the decode table, wherein the physical block corresponding to the logical block to be read is determined to be the new block if the data read from the physical block corresponding to the logical block to be read is the same as the data recorded in the decode table.

9. The data accessing method according to claim **3**, further comprising recording the indictor in a logical-physical mapping table.

10. The data accessing method according to claim **3**, further comprising recording the indicator in a redundant area of each of the physical blocks.

11. A controller, suitable for a flash memory storage device having a flash memory, wherein the flash memory comprises a plurality of physical blocks, and the physical blocks are grouped into at least a data area and a spare area, the controller comprising:

a microprocessor unit, for obtaining a logical block to be read and a page to be read from a read command received from a host;

a data perturbation module, electrically connected to the microprocessor unit for decoding data read from the flash memory;

a flash memory interface, electrically connected to the microprocessor unit for accessing the flash memory;

a buffer memory, electrically connected to the microprocessor unit for temporarily storing data; and

a memory management module, electrically connected to the microprocessor unit for determining whether a physical block in the data area corresponding to the logical block to be read is a new block and transmitting a predetermined data to the host when the physical block corresponding to the logical block to be read is the new block.

12. The controller according to claim **11**, wherein the predetermined data is 0x00 or 0xFF.

13. The controller according to claim **11**, wherein the memory management module records an indicator for each of the physical blocks to indicate that the physical block is the new block during a card activation process performed to the flash memory storage device, and the memory management module determines whether the physical block in the data area corresponding to the logical block to be read is the new block according to the indicator.

14. The controller according to claim **13**, wherein the memory management module writes the predetermined data into a predetermined page of each of the physical blocks during the card activation process, and the memory management module reads data from the predetermined page of the physical block corresponding to the logical block to be read when the physical block corresponding to the logical block to be read is the new block.

15. The controller according to claim **14**, wherein the predetermined page is a first page of the physical block.

16. The controller according to claim **13**, wherein the memory management module writes the predetermined data into all the pages of one of the physical blocks in the spare area during the card activation process, and the memory management module reads data from a page corresponding to the page to be read in the one of the physical blocks when the physical block corresponding to the logical block to be read is the new block.

17. The controller according to claim **13**, wherein the memory management module changes the data read from the physical block corresponding to the logical block to be read into the predetermined data when the physical block corresponding to the logical block to be read is the new block.

18. The controller according to claim **11**, wherein the memory management module decodes the data 0xFF in each of the physical blocks by using the data perturbation module and records the decoded data 0xFF into a decode table during the card activation process, and the memory management module determines the physical block corresponding to the logical block to be read to be the new block when the data read from the physical block corresponding to the logical block to be read is the same as the data recorded in the decode table.

19. The controller according to claim **13**, wherein the memory management module records the indicator in a logical-physical mapping table.

20. The controller according to claim **13**, wherein the memory management module records the indicator in a redundant area of each of the physical blocks.

21. The controller according to claim **11**, wherein the flash memory storage device is a flash drive, a memory card, or a solid state drive (SSD).

22. A flash memory storage system, comprising:

a flash memory, for storing data, wherein the flash memory comprises a plurality of physical blocks, and the physical blocks are grouped into at least a data area and a spare area;

a connector, for connecting to a host; and

a controller, electrically connected to the flash memory and the connector, the controller obtaining a logical block to be read and a page to be read from a read command received from a host, determining whether a physical block in the data area corresponding to the logical block to be read is a new block, and transmitting a predetermined data to the host when the physical block corresponding to the logical block to be read is the new block,

wherein the controller has a data perturbation module for decoding data read from the flash memory.

23. The flash memory storage system according to claim **22**, wherein the predetermined data is 0x00 or 0xFF.

24. The flash memory storage system according to claim **22**, wherein the controller records an indictor for each of the physical blocks to indicate that the physical block is the new block during a card activation process, and the controller determines the physical block in the data area corresponding to the logical block to be read to be the new block according to the indicator.

25. The flash memory storage system according to claim **24**, wherein the controller writes the predetermined data into a predetermined page of each of the physical blocks during the card activation process, and the controller reads data from the predetermined page of the physical block corresponding to the logical block to be read when the physical block corresponding to the logical block to be read is the new block.

* * * * *

CERTIFICATE OF SERVICE AND FILING

I certify that I electronically filed the foregoing document using the Court's CM/ECF filing system on July 6, 2015. All counsel of record were served via CM/ECF on July 6, 2015.

/s/ David M. Barkan

David M. Barkan
Fish & Richardson P.C.
500 Arguello Street
Redwood City, CA 94063
(650) 839-5070

Attorneys for Appellant,
PHISON ELECTRONICS CORP.

CERTIFICATE OF COMPLIANCE

The undersigned attorney certifies that Phison Electronics Corp.'s Opening Brief complies with the type-volume limitation set forth in Fed. R. App. P. 32(a)(7)(B). The relevant portions of the brief, including all footnotes, contain 8,187 words, as determined by Microsoft Word.

Dated: July 6, 2015

/s/ David M. Barkan

David M. Barkan
Fish & Richardson P.C.
500 Arguello Street
Redwood City, CA 94063
(650) 839-5070

Attorneys for Appellant,
PHISON ELECTRONICS CORP.